# Aruba Mobility Controller and Access Point Series Security Target

Version 1.0
09/29/2014

**Prepared for:**
**Aruba Networks, Inc.**

1344 Crossman Avenue
Sunnyvale, CA 94089-1113

**Prepared By:**

***Leidos***

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive, Columbia, Maryland 21046

**1. SECURITY TARGET INTRODUCTION ..... 4**

1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION ..... 4
1.2 CONFORMANCE CLAIMS ..... 7
1.3 CONVENTIONS ..... 7
*1.3.1 Acronyms ..... 7*

**2. TOE DESCRIPTION ..... 9**

2.1 TOE OVERVIEW ..... 10
2.2 TOE ARCHITECTURE ..... 10
*2.2.1 Physical Boundaries ..... 12*
*2.2.2 Logical Boundaries ..... 13*
2.3 TOE DOCUMENTATION ..... 15

**3. SECURITY PROBLEM DEFINITION ..... 17**

3.1 ORGANIZATIONAL POLICIES ..... 17
3.2 THREATS ..... 17
3.3 ASSUMPTIONS ..... 18

**4. SECURITY OBJECTIVES ..... 19**

4.1 SECURITY OBJECTIVES FOR THE TOE ..... 19
4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT ..... 20

**5. IT SECURITY REQUIREMENTS ..... 21**

5.1 EXTENDED REQUIREMENT DEFINITIONS ..... 21
5.2 TOE SECURITY FUNCTIONAL REQUIREMENTS ..... 21
*5.2.1 Security audit (FAU) ..... 23*
*5.2.2 Cryptographic support (FCS) ..... 30*
*5.2.3 User data protection (FDP) ..... 42*
*5.2.4 Identification and authentication (FIA) ..... 43*
*5.2.5 Security management (FMT) ..... 48*
*5.2.6 Protection of the TSF (FPT) ..... 50*
*5.2.7 Resource utilisation (FRU) ..... 52*
*5.2.8 TOE access (FTA) ..... 52*
*5.2.9 Trusted path/channels (FTP) ..... 54*
5.3 TOE SECURITY ASSURANCE REQUIREMENTS ..... 55
*5.3.1 Development (ADV) ..... 56*
*5.3.2 Guidance documents (AGD) ..... 56*
*5.3.3 Life-cycle support (ALC) ..... 58*
*5.3.4 Tests (ATE) ..... 58*
*5.3.5 Vulnerability assessment (AVA) ..... 59*

**6. TOE SUMMARY SPECIFICATION ..... 61**

6.1 SECURITY AUDIT ..... 61
6.2 CRYPTOGRAPHIC SUPPORT ..... 63
6.3 USER DATA PROTECTION ..... 71
6.4 IDENTIFICATION AND AUTHENTICATION ..... 71
6.5 SECURITY MANAGEMENT ..... 74
6.6 PROTECTION OF THE TSF ..... 75
6.7 RESOURCE UTILIZATION ..... 76
6.8 TOE ACCESS ..... 76
6.9 TRUSTED PATH/CHANNELS ..... 77

**7. PROTECTION PROFILE CLAIMS ..... 79**

**8. RATIONALE ..... 80**

8.1 SECURITY OBJECTIVES RATIONALE ..... 80

*8.1.1 Security Objectives Rationale for the TOE and Environment* ..... *80*
8.2 SECURITY REQUIREMENTS RATIONALE ..... 84
*8.2.1 Security Functional Requirements Rationale* ..... *84*
8.3 SECURITY ASSURANCE REQUIREMENTS RATIONALE ..... 91
8.4 REQUIREMENT DEPENDENCY RATIONALE ..... 91
8.5 TOE SUMMARY SPECIFICATION RATIONALE ..... 92

**LIST OF TABLES**

**Table 1 TOE Security Functional Components** ..... 23
**Table 2 Audit Events** ..... 27
**Table 3 EAL 1 Assurance Components** ..... 55
**Table 4 Cryptographic Functions** ..... 64
**Table 5 NIST SP800-56A Conformance** ..... 64
**Table 6 NIST SP800-56B Conformance** ..... 65
**Table 7 Environment to Objective Correspondence** ..... 80
**Table 8 Objective to Requirement Correspondence** ..... 85
**Table 9 Requirement Dependencies** ..... 92
**Table 10 Security Functions vs. Requirements Mapping** ..... 93

# 1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is a Wireless Local Area Network (WLAN) access system comprising Aruba Mobility Controllers and Access Points (both with an embedded ArubaOS). The Aruba Mobility Controllers are wireless switch appliances that provide a wide range of wireless and wired network mobility, security, centralized management, auditing, authentication, and remote access. The Aruba Access Point appliances service wireless clients[1] and can monitor radio frequency spectrums to detect intrusions, denial of service (DoS) attacks, and other vulnerabilities. The ArubaOS is a suite of mobility applications that runs on all Aruba controllers and APs and allows administrators to configure and manage the wireless and mobile user environment.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

## 1.1 Security Target, TOE and CC Identification

**ST Title –** Aruba Mobility Controller and Access Point Series Security Target

**ST Version** – Version 0.10

**ST Date** – 2/5/2014

**TOE Identification** – Aruba Mobility Controller and Access Point Series, (ArubaOS version 6.3.1.5-FIPS – see table below.

[1] The wireless client is part of the IT environment.

| Product | Part Number(s) | Required Software Licenses | Firmware Version |
|---|---|---|---|
| Aruba 7210 Mobility Controller (FIPS) | • 7210-F1<br>• 7210-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |
| Aruba 7220 Mobility Controller (FIPS) | • 7220-F1<br>• 7220-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |
| Aruba 7240 Mobility Controller (FIPS) | • 7240-F1<br>• 7240-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |
| Aruba 6000 Mobility Controller (FIPS) | • 6000-400-F1<br>• 6000-400-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |
| Aruba 3200 Mobility Controller (FIPS) | • 3200-F1<br>• 3200-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |
| Aruba 3400 Mobility Controller (FIPS) | • 3400-F1<br>• 3400-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |
| Aruba 3600 Mobility Controller (FIPS) | • 3600-F1<br>• 3600-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |
| Aruba 650 Branch Office Controller (FIPS) | • 650-F1<br>• 650-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |

| | | | |
|---|---|---|---|
| Aruba 620 Branch Office Controller (FIPS) | • 620-F1<br>• 620-USF1 | • Policy Enforcement Firewall<br>• RFprotect<br>• Advanced Cryptography | 6.3.1.5-FIPS |
| AP-92 Access Point | • AP-92-F1 | N/A | 6.3.1.5-FIPS |
| AP-93 Access Point | • AP-93-F1 | N/A | 6.3.1.5-FIPS |
| AP-104 Access Point | • AP-104-F1 | N/A | 6.3.1.5-FIPS |
| AP-105 Access Point | • AP-105-F1 | N/A | 6.3.1.5-FIPS |
| AP-114 Access Point | • AP-114-F1 | N/A | 6.3.1.5-FIPS |
| AP-115 Access Point | • AP-115-F1 | N/A | 6.3.1.5-FIPS |
| AP-134 Access Point | • AP-134-F1 | N/A | 6.3.1.5-FIPS |
| AP-135 Access Point | • AP-135-F1 | N/A | 6.3.1.5-FIPS |
| AP-175 Access Point | • AP-175AC-F1<br>• AP-175DC-F1<br>• AP-175P-F1 | N/A | 6.3.1.5-FIPS |
| AP-224 Access Point | • AP-224-F1 | N/A | 6.3.1.5-FIPS |
| AP-225 Access Point | • AP-225-F1 | N/A | 6.3.1.5-FIPS |
| RAP-3WN Access Point | • RAP-3WN-F1<br>• RAP-3WN-USF1<br>• RAP-3WNP-F1<br>• RAP-3WNP-USF1 | N/A | 6.3.1.5-FIPS |
| RAP-5WN Remote Access Point | • RAP-5WN-F1 | N/A | 6.3.1.5-FIPS |
| RAP-108 Remote Access Point | • RAP-108-F1<br>• RAP-108-USF1 | N/A | 6.3.1.5-FIPS |
| RAP-109 Remote Access Point | • RAP-109-F1<br>• RAP-109-USF1 | N/A | 6.3.1.5-FIPS |
| RAP-155 Remote Access Point | • RAP-155-F1<br>• RAP-155-USF1<br>• RAP-155P-F1<br>• RAP-155P-USF1 | N/A | 6.3.1.5-FIPS |

**TOE Developer** – Aruba Networks, Inc.

**Evaluation Sponsor** – Aruba Networks, Inc.

**CC Identification** – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3, July 2009

## 1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- *Protection Profile for Wireless Local Area Network (WLAN) Access Systems*, version 1.0, 01 December 2011 (WLASPP)
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 3, July 2009.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 3, July 2009.
    - Part 3 Conformant
    - Assurance Level: EAL 1

## 1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
    - Extended Requirements are allowed to create requirements should the Common Criteria not offer suitable requirements to meet the ST needs. To ensure these requirements are explicitly identified, the ending "_EXT" is appended to the newly created short name and the component.
- The WLASPP uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

### 1.3.1 Acronyms

| | |
|---|---|
| AAA | Authentication, Authorization, and Accounting |
| AES | Advanced Encryption Standard |
| AP | Access Point |
| BOC | Branch Office Controller |
| CC | Common Criteria |

| | |
|---|---|
| CLI | Command Line Interface |
| CP | Control Plane |
| DP | Data Plane |
| DoS | Denial of Service |
| EAP | Extensible Authentication Protocol |
| FP | Fast Path |
| FPGA | Field Programmable Gate Array |
| FIPS | Federal Information Processing Standard |
| GRE | Generic Routing Encapsulation |
| GUI | Graphical User Interface |
| IKE | Internet Key Exchange |
| IPsec | Internet Protocol Security |
| LDAP | Lightweight Directory Access Protocol |
| MAC | Medium Access Control |
| MC | Mobility Controller |
| HMAC-MD5 | Hashed Message Authentication Code – Message Digest 5 |
| NAT | Network Address Translation |
| NTP | Network Time Protocol |
| PAPI | Programming Application Program Interface |
| PP | Protection Profile |
| RADIUS | Remote Authentication Dial In User Service |
| RF | Radio Frequency |
| RNG | Random Number Generator |
| SP | Slow Path |
| SSH | Secure Shell |
| TACACS+ | Terminal Access Controller Access-Control System + |
| TLS | Transport Layer Security |
| TOE | Target of Evaluation |
| TSF | TOE Security Function |
| VLAN | Virtual Local Area Network |
| VPN | Virtual Private Network |
| WIDS | Wireless Intrusion Detection System |
| WIP | Wireless Intrusion Protection |
| WLAN | Wireless Local Area Network |
| WPA | Wi-Fi Protected Access |

# 2. TOE Description

The Target of Evaluation (TOE) consists of Aruba Mobility Controller appliances and access points, running ArubaOS v6.3.1.5-FIPS.

The TOE is a Wireless Local Area Network (WLAN) access system comprising Aruba Mobility Controllers, Access Points, and the ArubaOS. The WLAN PP defines this technology type as "one or more components that provide secure wireless access to a wired or wireless network". The Aruba Mobility Controllers are wireless switch appliances that provide a wide range of security services and features including wireless and wired network mobility, security, centralized management, auditing, authentication, and remote access. The Aruba Access Point appliances service wireless clients[2] and can monitor radio frequency spectrums to detect intrusions, denial of service (DoS) attacks, and other vulnerabilities. The ArubaOS is a suite of mobility applications that runs on all Aruba controllers and APs, and allows administrators to configure and manage the wireless and mobile user environment. Figure 1 shows an example of a WLAN Access System environment configuration[3]. Figure 2 shows an example of a WLAN Access System configuration. This configuration includes one AP and one MC. This should not be misconstrued as the only configuration as multiple MCs and APs can comprise the TOE. However, this is the minimum configuration required in the CC mode. The rest of this section will describe, at a high-level, an overview of the TOE architecture, define the scope of evaluation and the physical boundary of the TOE, and summarize the security functionality provided by the TOE.

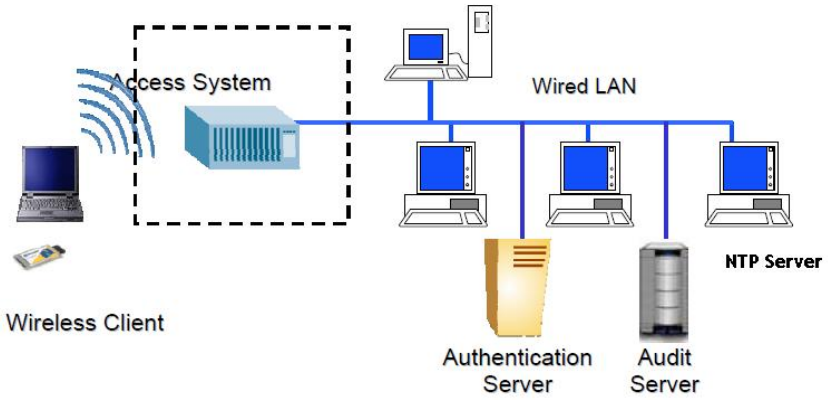


**Figure 1: Example of WLAN Access System Environment**

**Figure 2: Example of WLAN Access System**

The AP is connected to the Controller via wired Ethernet Local Area Network (LAN) over an IP network or wired directly to the Controller. The control data passed over this connection is protected using IPsec based on a FIPS approved cryptographic module. The AP and MC use GRE as the tunneling protocol to encapsulate IEEE 802.11

[2] Wireless client is not part of the TOE

[3] Other wireless configurations may exist and still meet requirements identified in the PP. In all cases, wireless traffic must be able to pass to the wired network via the wireless access system providing the necessary security.

traffic (data from wireless clients) over the IP wired network. As a result, APs can be distributed as necessary and need not be kept in close proximity with a physically secure connection to the associated Controller.

In an encrypted WLAN, a wireless client first associates with an AP and then authenticates (IEEE 802.11i[4]) using credentials to obtain access to the network (an IP address) and establish a session with the TOE. The authenticated wireless client is then assigned a role based on the configuration in the Mobility Controller.

Each authenticated wireless client can also be placed into a VLAN. While all authenticated wireless clients can be placed into a single VLAN, the TOE (Mobility Controller) allows administrators to group wireless clients into separate VLANs. This enables separation and isolation of groups of wireless clients and their access to network resources. For example, administrators can place authorized employee clients into one VLAN and temporal clients, such as contractors or guests, into a separate VLAN.

## 2.1 TOE Overview

The TOE consists of the following components:

- Aruba Mobility Controllers
- Aruba Access Points
- ArubaOS.

In the CC evaluated configuration, the TOE (all components that make up the WLAN access system—at a minimum, one Controller and one AP) must be configured to operate in the FIPS 140-2 Approved mode of operation. In FIPS-Approved mode, various weak protocols and algorithms are disabled. Please reference the appropriate FIPS 140-2 Security Policy documents for each controller and access point for more details at http://csrc.nist.gov/groups/STM/cmvp/index.html.

## 2.2 TOE Architecture

At a high level, Aruba Mobility Controllers are hardware appliances consisting of a multicore network processor, Ethernet interfaces, and required supporting circuitry and power supplies enclosed in a metal chassis. The software running on the Mobility Controller is called ArubaOS, which consists of two main components, both implemented on multiple cores within a single network processor:

- Control Plane (CP)—implements functions which can be handled at lower speeds such as Mobility Controller system management (CLI and Web GUI), user authentication (e.g. 802.1X, RADIUS, LDAP), Internet Key Exchange (IKE), auditing/logging (syslog), Wireless IDS (WIDS), and termination of protocols operating at the system level (e.g. SSH, TLS, NTP, etc.). The Control Plane runs the Linux operating system along with various user-space applications (described below).
- Data Plane (DP)—implements functions that must be handled at high speeds such as high-speed switching functions (forwarding, VLAN tagging/enforcement, bridging), termination of 802.11 associations/sessions, tunnel termination (GRE, IPsec), stateful firewall and deep packet inspection functions, and cryptographic acceleration. The Data Plane runs a lightweight, proprietary real-time OS which is known as “SOS” (an acronym whose definition is no longer known).

The Control Plane and Data Plane are inseparable. Administrators install the software by loading a single file, identified as “ArubaOS”. Internally, the Mobility Controller unpacks the ArubaOS software image into its various components. A given ArubaOS software image has a single version number, and includes all software components necessary to operate both mobility controllers and APs. The mobility controller is responsible for storing the ArubaOS components needed to operate the APs, allowing APs to download their operating software from the mobility controller.

The CP runs the Linux OS, along with various custom user-space applications which provide the following CP functions:

- Monitors and manages critical system resources, including processes, memory, and flash

[4] Implements 802.1X for wireless access points to address the security vulnerabilities found in WEP.

- Sends and receives IPsec-encapsulated PAPI[5] protocol messages to and from managed APs as well as other mobility controllers
- Manages system configuration and licensing
- Manages an internal database used to store licenses, user authentication information, etc
- Provides network anomaly detection, hardware monitoring, mobility management, wireless management, and radio frequency management services
- Provides a Command Line Interface (CLI)
- Provides a web-based (HTTPS/TLS) management UI for the mobility controller
- Provides various WLAN station and AP management functions
- Provides authentication services for the system management interfaces (CLI, web GUI) as well as for WLAN users
- Provides IPsec key management services for APs, VPN users, and connections with other Aruba mobility controllers
- Provides network time protocol service for APs, point to point tunneling protocol services for users, layer 2 tunneling protocol services for users, , , SSH services for incoming management connections, SNMP client/agent services, and protocol independent multicast (routing) services for the controller
- Provides syslog services by sending logs to the operating environment.

The Linux OS running on the CP is a standard unmodified 2.6.32 kernel. Linux is a soft real-time, multi-threaded operating system that supports memory protection between processes. Only Aruba provided interfaces are used, and the CLI is a restricted command set. Administrators do not have access to the Linux command shell or operating system.

The DP is further subdivided into two subcomponents: Fast Path (FP) and Slow[6] Path (SP). The FP implements high-speed packet forwarding based on various proprietary tables and sends the packets to SP. The SP manages (create, delete, and age entries) all DP tables such as user, stateful firewall rules, station, tunnel, route, ARP cache, session, bridge, VLAN[7], and port. The SP also performs deep packet inspection and cryptographic processing.

The data plane is implemented on a multi-core network processor[8]. There is a lightweight, Aruba-proprietary OS running on the network processor called SOS. SOS contains an Ethernet driver, a serial driver, a logging facility, semaphore support, and a crypto driver. This OS is not a general purpose operating system. In the Aruba 6000 with M3 controller card, an FPGA is also used to control and monitor the switch fabric, Ethernet interface hardware, and provide security functionality such as filtering.

The DP and CP run on different hardware platforms but the security functionality remains the same, regardless of the model. The differences in the platforms are in the processors, memory capacity, physical interfaces, FPGA implementation, etc., and are based on performance and scalability requirements. The table below shows the different models based on maximum number of APs and users supported.

[5] PAPI is an Aruba-proprietary WLAN management protocol and provides no direct security.
[6] The entire DP (including both FP and SP elements) is a high-speed packet processor, so the SP designation should be understood to be relative.
[7] A VLAN has the same attributes as a physical LAN, but it allows for end devices to be grouped together even if they are not located on the same network switch. Network reconfiguration can be done through the Aruba software instead of physically relocating devices.

| Product | Max. # of APs | Max. # of Users | Typical Deployment |
|---|---|---|---|
| Aruba 7200 Series | 2,048 | 32,768 | Headquarters/ Large Campus |
| Aruba 6000/M3 | 512 | 8,192 | Headquarters/ Large Campus |
| Aruba 3000 Series | 128 | 2,048 | Medium/Large Enterprise/Campus |
| Aruba 620/650 | 16 | 256 | Branch Office |

The Aruba AP is a hardware device that is enclosed in a plastic or metal casing. All APs contain chips to provide IEEE 802.11 wireless LAN functionality. Some models contain a separate CPU, while other models combine the CPU with the wireless LAN chip (an integrated approach known as "system on a chip"). Some AP models contain integrated antennas, while other models provide connectors for attaching external antennas. Software functionality for the APs is provided by ArubaOS, which is downloaded from the mobility controller and stored in a local flash memory partition. In the case of the APs, ArubaOS consists of a Linux kernel and various custom user-space applications. Although the AP's operating system is named ArubaOS, the Linux kernel and user-space applications are different from those running on the mobility controller. The version number of ArubaOS running on the AP and the version number of ArubaOS running on the controller are the same; the two software images are bundled into a single image file that is installed by the administrator on the mobility controller. Similar to the controllers, the security functionality of the different models is the same with differences in platforms based on performance and scalability requirements only. At a high level, Aruba Access Points consist of the following subsystems:

- Processor subsystem—performs the packet processing functions on the packet.
- Memory subsystem—contains memory which supports the Processor subsystem.
- Ethernet Controller (i.e., Network Interface Controller) subsystem—includes integrated Ethernet Media Access Control (MAC) for transfer of 10/100 Ethernet packets between the AP and the wired network.
- Radio Controller subsystem—there are one or two (depending on model) radio controllers, 802.11a/n (5 GHz range) and 802.11b/g/n (2.4 GHz range).
- Wireless Antenna subsystem—interface between the wireless world and the AP. The antenna handles both 5 GHz and 2.4 GHz ranges. Some AP models include connectors for external antennas, while other AP models contain integrated antennas.
- PoE (Power over Ethernet) subsystem—receives 48V power over the Ethernet.
- USB subsystem—the AP-70 and RAP-5wn support one USB V2.0 compliant port (up to 480 Mbps). A PCI to USB 2.0 controller is used to interface to the system host.
- Serial subsystem—all 802.11n APs support a serial console port that utilizes a RJ45 jack and connects directly to serial port 0 via the RS232 transceiver.

Aruba APs may or may not perform cryptographic processing, depending on administrator configuration. The default mode of operation is known as "tunnel mode", in which raw encrypted 802.11 frames are passed through the AP and processed by the Mobility Controller without decryption or further processing in between. This mode of operation places fewer security constraints on the AP, since cleartext network traffic is never present in the AP. Other modes of operation are available as well, including "decrypt-tunnel mode", in which wireless traffic is decrypted by the AP and forwarded to the Mobility Controller, and "bridge mode", in which wireless traffic is decrypted and forwarded directly from the AP to the local LAN segment. In the CC-evaluated configuration, only tunnel mode is used.

### 2.2.1 Physical Boundaries

The TOE consists of the following components:

- Aruba Mobility Controllers: Aruba 620, 650, 3200, 3400, 3600, 6000, 7210, 7220, and 7240.
- Aruba Access Points: Aruba AP-92, AP-93, AP-104, AP-105, AP-114, AP-115, AP-134, AP-135, AP-175, AP-224, AP-225, RAP-3WN, RAP-5WN, RAP-108, RAP-109, and RAP-155.

- ArubaOS version 6.3.1.5-FIPS

The differences in the models include the number of ports, interfaces, throughput and processing speed, memory and storage. Although these models have different specifications (in terms of performance and capabilities), they all provide the same security functions described in the ST; therefore, they have been considered to be the same for the purposes of the ST description. There is no difference between the products and the TOE. Since the TOE is a WLAN access system, the physical boundary of each product that comprises the WLAN is the hard steel or plastic encasing.

The ArubaOS consists of a base software package with add-on software modules that can be activated by installing the appropriate license key. Three SFR-enforcing software modules are required to be licensed and installed in the CC evaluated configuration. The base ArubaOS software includes the following functions:

- Centralized configuration and management of APs
- Wireless client authentication to an external authentication server or to the controller's internal database
- Encryption
- Mobility with fast roaming
- RF management and analysis tools.

The following table summarizes the required software modules.

| Required Software Module | Description |
|---|---|
| Policy Enforcement Firewall | Provides identity-based security for wired and wireless clients. Stateful firewall enables classification based on client identity, device type, location, and time of day, and provides differentiated access for different classes of users. |
| RFprotect | Detects, classifies and limits designated wireless security threats such as rogue APs, DoS attacks, malicious wireless attacks, impersonations, and unauthorized intrusions. Eliminates need for separate system of RF sensors and security appliances. Also provides spectrum intelligence and spectrum visibility when used with compatible AP platforms. |
| Advanced Cryptography | Required for SuiteB, AES-GCM and ECDSA functionality. |

The wireless client can be any device that uses a wireless network interface that is Wi-Fi Certified. Specifically, it must be WPA2 compliant to support the cryptographic and authentication (e.g., certificate) requirements of the TOE. Note that WPA2 compliance is a specific subset of the Wi-Fi certification. For more information, please see http://certifications.wi-fi.org/wbcs_certified_products.php?lang=en

The TOE relies on third-party software and hardware components in the operating environment. The TOE can utilize an external audit server (support syslog) to store audit records and external authentication server (support RADIUS, LDAP, TACACS+) to authenticate users. In addition, the TOE uses an external Time server (support NTP) to obtain reliable time stamps and external SNMP server to capture SNMP traps. For security reasons, only SNMPv3 is allowed in the evaluated configuration. The remote administrator can use a web browser (supported browsers: Microsoft Internet Explorer 8.x on Windows XP, Windows Vista, Windows 7, and MacOS; Mozilla Firefox 3.x on Windows XP, Windows Vista, Windows 7, and MacOS; Apple Safari 5.x on MacOS) to access the Web GUI interface and/or use SSH client to access the CLI. The local administrator can use the serial port to access the CLI. Neither the web browser or SSH client is part of the TOE. Note that Telnet cannot be used to access the CLI in the CC evaluated configuration.

### 2.2.2 Logical Boundaries

This section summarizes the security functions provided by Mobility Controller:

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Resource utilisation
- TOE access
- Trusted path/channels

The TOE protects itself from tampering and bypass through several mechanisms implemented by the TOE and the operating environment. The underlying operating system separates processes into separate domains and prevents one process from accessing memory space of another process. The operating system is non-modifiable and the interfaces are strictly limited. The TOE relies on physical security to protect data transmitted between the TOE components from unauthorized modification. Remote administration used by administrators to manage the TOE is secured through TLS (Web GUI) or SSH (CLI). All administrators must be identified and authenticated either by the TOE or an external authentication server. Inactive sessions are terminated after an administrator-specified time period. In addition to authentication, the TOE also verifies that users are authorized to perform management functions based on their roles. An internal real-time clock chip and/or external NTP server provide the reliable time source and external syslog server stores and protects the audit trail from tampering.

The sections below summarize the security functions provided by the TOE.

#### 2.2.2.1 Security audit

The TOE is capable of auditing security relevant events such as logins, administrator actions, use of trusted channel and path, cryptographic operations, resource limitation exceeded, etc. Each audit event includes the date and time of the event, the type of event, the subject identity (if applicable), and the outcome of the event. The administrator can include and exclude events to be audited based on specific criteria.

The TOE may utilize its internal real-time clock chip and/or an extenral NTP server to provide a reliable timestamp and syslog server to store and protect the audit trail. The administrator is provided an interface in the operating environment to read audit logs and that interface is restricted.

#### 2.2.2.2 Cryptographic support

The TOE has been certified as a FIPS 140-2 cryptographic module (FIPS 140-2 certified Certificates #:1297, #1116, #1109, #1077, #1075, #1727, #1838, #1828, and #1815). The FIPS overall level is 2.The logical interfaces used for the input and output of plaintext cryptographic key components, authentication data, and CSPs are logically separated from all other interfaces using a trusted path where "trusted path" is interpreted to include a communications channel established using a FIPS 140-2 Level 2 cryptographic module and the HTTPS protocol between the cryptomodule and the external IT entity. The cryptographic module only employs FIPS-Approved RNG, key generation, establishment, zeroization, encryption, digital signature, and hashing algorithms as specified by the FCS requirements.

#### 2.2.2.3 User data protection

The TOE ensures that any data packets passing through do not inadvertently contain any residual information that might be disclosed inappropriately.

#### 2.2.2.4 Identification and authentication

The TOE can maintain administrator and user attributes, including credentials such as username and password for administrators and session key and role for remote authenticated users (username and password are stored in the

internal database or authentication server). The TOE requires identification and authentication (either locally or remotely through external authentication server, internally, or both) of administrators managing the TOE. Wireless clients are identified and authenticated by different authentication mechanisms such as 802.1X, etc. More detailed information is provided in section 6.1.4. After an administrator-specified number of failed attempts, the user account is locked out. In addition, the password mechanism can be configured to have a minimum length of six characters.

#### 2.2.2.5 Security management

The TOE provides the capability to manage auditing, cryptographic operations, password minimum length enforcement, user accounts, advisory banner, and timeout (inactivity threshold) value. The management functions are restricted to an administrator role. The role must have the appropriate access privileges or access will be denied. The wireless user role has no access to the management interfaces. The FIPS-certified TOE ensures that only secure values are accepted for security attributes.

#### 2.2.2.6 Protection of the TSF

The TOE provides integrity and security protection for all communication between its components. This prevents unauthorized modification or disclosure of TSF data during transmission. The TOE also protects itself against replay attacks using cryptographic protocols.

The TOE provides self-tests to ensure the correct operation of the cryptographic functions and TSF hardware. There is an option for the administrator to verify the integrity of stored TSF executable code.

The TOE includes mechanisms so that the administrator can determine the TOE version and update the TOE securely using digital signatures and published hashes.

#### 2.2.2.7 Resource utilization

The TOE can enforce maximum usage quotas on the number of concurrent sessions available to a defined group of users (role).

#### 2.2.2.8 TOE access

The TOE allows administrators to configure a period of inactivity for administrator and wireless user sessions. Once that time period has been reached while the session has no activity, the session is terminated. Administrators as well as wireless users can also terminate their own sessions at any time. A warning banner is displayed at the management interfaces (Web GUI and CLI) to advise users on appropriate use and penalty for misuse of system.

The TOE can restrict the ability to connect to administrative interfaces based on time/date, location, and device MAC address and blacklist status.

#### 2.2.2.9 Trusted path/channels

The TOE provides an encrypted channel between itself and third-party trusted IT entities in the operating environment. The TOE also provides a protected communication path between itself and wireless users.

## 2.3 TOE Documentation

Aruba Networks offers a series of documents that describe the installation and configuration of Mobility Controllers and Access Points as well as guidance for subsequent use and administration of the applicable security features. The documentation is available online at http://support.arubanetworks.com. The following documents are referenced throughout this ST:

| | |
|---|---|
| [USER] | ArubaOS 6.3.x User Guide, Ref 0511497-00 |
| [CLI] | ArubaOS 6.3.x Command Line Interface, Ref 0511500-00 |
| [SYSLOG] | ArubaOS 6.3.x Syslog Messages Guide, Ref 0511324-01 |
| [MIB] | ArubaOS 6.3.x MIB Reference Guide, Ref 0511323-01 |
| [RN] | ArubaOS 6.3.1.5 Release Notes, Ref 0511467-05v1 |
| [FIPS] | ArubaOS 6.3 FIPS Security Policy (available at CMVP website) |
| [QUICK] | ArubaOS 6.3 Quick Start Guide, Ref 0511320-02 |

# 3. Security Problem Definition

The Security Problem Definition (composed of organizational policies, threat statements, and assumption) has been drawn from the *Protection Profile for Wireless Local Area Network (WLAN) Access Systems*, version 1.0, 01 December 2011 (WLASPP). The WLASPP offers additional information about the identified threats, but that has not been reproduced here and the WLASPP should be consulted if there is interest in that material.

In general, the WLASPP has presented a Security Problem Definition appropriate for network infrastructure devices and as such is applicable to the Mobility Controller and Access Point Series TOE.

## 3.1 Organizational Policies

**P.ACCESS_BANNER**

The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

**P.ACCOUNTABILITY**

The authorized users of the TOE shall be held accountable for their actions within the TOE.

**P.ADMIN_ACCESS**

Administrators shall be able to administer the TOE both locally and remotely through protected communications channels.

**P.COMPATIBILITY**

The TOE must meet Request for Comments (RFC) requirements for implemented protocols to facilitate inter-operation with other network equipment (e.g., certificate authority, NTP server) using the same protocols.

**P.EXTERNAL_SERVERS**

The TOE must support standardized (RFCs) protocols for communication with a centralized audit server and a RADIUS authentication server.

## 3.2 Threats

**T.ADMIN_ERROR**

An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.

**T.RESOURCE_EXHAUSTION**

A process or user may deny access to TOE services by exhausting critical resources on the TOE.

**T.TSF_FAILURE**

Security mechanisms of the TOE may fail, leading to a compromise of the TSF.

**T.UNAUTHORIZED_ACCESS**

A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.

**T.UNAUTHORIZED_UPDATE**

A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.

**T.UNDETECTED_ACTIONS**

Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.

**T.USER_DATA_REUSE**

User data may be inadvertently sent to a destination not intended by the original sender.

## 3.3 Assumptions

**A.NO_GENERAL_PURPOSE**

It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of the TOE.

**A.NO_TOE_BYPASS**

Information cannot flow between the wireless client and the internal wired network without passing through the TOE.

**A.PHYSICAL**

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

**A.TRUSTED_ADMIN**

TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

# 4. Security Objectives

Like the Security Problem Definition, the Security Objectives have been drawn from the *Protection Profile for Wireless Local Area Network (WLAN) Access Systems*, version 1.0, 01 December 2011 (WLASPP). The WLASPP offers additional information about the identified security objectives, but that has not been reproduced here and the WLASPP should be consulted if there is interest in that material.

In general, the WLASPP has presented a Security Objectives appropriate for network infrastructure devices and as such are applicable to the Mobility Controller and Access Point Series TOE.

## 4.1 Security Objectives for the TOE

**O.AUTH_COMM**

The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity.

**O.CRYPTOGRAPHIC_FUNCTIONS**

The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.

**O.DISPLAY_BANNER**

The TOE will display an advisory warning regarding use of the TOE.

**O.FAIL_SECURE**

The TOE shall fail in a secure manner following failure of the power-on self tests.

**O.PROTECTED_COMMUNICATIONS**

The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.

**O.PROTOCOLS**

The TOE will ensure that standardized protocols are implemented in the TOE to RFC and/or Industry specifications to ensure interoperability, that also support communication with a centralized audit server and a RADIUS authentication server.

**O.REPLAY_DETECTION**

The TOE will provide a means to detect and reject the replay of authentication data and other TSF data and security attributes.

**O.RESIDUAL_INFORMATION_CLEARING**

The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.

**O.RESOURCE_AVAILABILITY**

The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (e.g., persistent storage).

**O.ROBUST_TOE_ACCESS**

The TOE will provide mechanisms that control an administrator’s logical access to the TOE and to control administrative access from a wireless client.

**O.SESSION_LOCK**

The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.

**O.SYSTEM_MONITORING**

The TOE will provide the capability to generate audit data and send those data to an external IT entity.

**O.TIME_STAMPS**

The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these timestamps.

**O.TOE_ADMINISTRATION**

The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.

**O.TSF_SELF_TEST**

The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

**O.VERIFIABLE_UPDATES**

The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.

**O.WIRELESS_CLIENT_ACCESS**

The TOE will provide the capability to restrict a wireless client in connecting to the TOE.

## 4.2 Security Objectives for the Environment

**OE.NO_GENERAL_PURPOSE**

There are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of the TOE.

**OE.NO_TOE_BYPASS**

Information cannot flow between external and internal networks located in different enclaves without passing through the TOE.

**OE.PHYSICAL**

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the IT environment.

**OE.TRUSTED_ADMIN**

TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

# 5. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the Protection Profile (PP): *Protection Profile for Wireless Local Area Network (WLAN) Access Systems*, version 1.0, 01 December 2011 (WLASPP). The refinements and operations already performed in that PP are not identified (e.g., highlighted) here, rather the requirements have been copied from that PP and any residual operations have been completed herein. Of particular note, the WLASPP made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the WLASPP which includes all the SARs for EAL1 as defined in the CC. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the WLASPP that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL1 assurance requirements alone. As such, those assurance activities have been reproduced in this ST to ensure they are included within the scope of the evaluation effort.

## 5.1 Extended Requirement Definitions

All of the extended requirements in this ST have been drawn from the WLASPP. The WLASPP defines the following extended SFRs and since they are not redefined in this ST, the WLASPP should be consulted for more information in regard to those CC extensions.

- FAU_STG_EXT.1: External Audit Trail Storage
- FAU_STG_EXT.3: Action in Case of Loss of Audit Server Connectivity
- FCS_CKM_EXT.4: Cryptographic Key Zeroization
- FCS_HTTPS_EXT.1: Explicit: HTTPS
- FCS_IPSEC_EXT Extended: Internet Protocol Security (IPsec) Communications
- FCS_RBG_EXT.1 Extended: Cryptographic operation (Random Bit Generation)
- FCS_SSH_EXT.1: Explicit: SSH
- FCS_TLS_EXT.1: Explicit: TLS
- FIA_PMG_EXT.1: Password Management
- FIA_UIA_EXT.1 User Identification and Authentication
- FIA_UAU_EXT.5 Extended: Password-based Authentication Mechanisms
- FIA_8021X_EXT.1 Extended: 802.1X Port Access Entity (Authenticator) Authentication
- FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition
- FIA_X509_EXT.1 Extended: X509 Certificates
- FPT_TST_EXT.1: TSF Testing
- FPT_TUD_EXT.1: Extended: Trusted Update
- FTA_SSL_EXT.1: TSF-initiated session locking

## 5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Mobility Controller and Access Point Series TOE.

| Requirement Class | Requirement Component |
|---|---|
| **FAU: Security audit** | FAU_GEN.1: Audit Data Generation |
| | FAU_GEN.2: User Audit Association |
| | FAU_SAR.1: Audit Review |
| | FAU_SAR.2: Restricted Audit Review |
| | FAU_SEL.1: Selective Audit |
| | FAU_STG.1: Protected Audit Trail Storage (Local Storage) |
| | FAU_STG_EXT.1: External Audit Trail Storage |
| | FAU_STG_EXT.3: Action in Case of Loss of Audit Server Connectivity |
| **FCS: Cryptographic support** | FCS_CKM.1(1): Cryptographic Key Generation (Symmetric Keys for WPA2 Connections) |
| | FCS_CKM.1(2): Cryptographic Key Generation (Asymmetric Keys) |
| | FCS_CKM.2(1): Cryptographic Key Distribution (PMK) |
| | FCS_CKM.2(2): Cryptographic Key Distribution (GTK) |
| | FCS_CKM_EXT.4: Cryptographic Key Zeroization |
| | FCS_COP.1(1): Cryptographic Operation (Data Encryption/Decryption) |
| | FCS_COP.1(2): Cryptographic Operation (Cryptographic Signature) |
| | FCS_COP.1(3): Cryptographic Operation (Cryptographic Hashing) |
| | FCS_COP.1(4): Cryptographic Operation (Keyed-Hash Message Authentication) |
| | FCS_COP.1(5): Cryptographic Operation (WPA2 Data Encryption/Decryption) |
| | FCS_HTTPS_EXT.1: Extended: HTTP Security (HTTPS) |
| | FCS_IPSEC_EXT.1: Extended: Internet Protocol Security (IPsec) Communications |
| | FCS_RBG_EXT.1: Extended: Cryptographic Operation: Random Bit Generation |
| | FCS_SSH_EXT.1: Extended: Secure Shell (SSH) |
| | FCS_TLS_EXT.1: Extended: Transport Layer Security (TLS) |
| **FDP: User data protection** | FDP_RIP.2: Full Resident Information Protection |
| **FIA: Identification and authentication** | FIA_8021X_EXT.1: Extended: 802.1X Port Access Entity (Authenticator) Authentication |
| | FIA_AFL.1: Authentication Failure Handling |
| | FIA_PMG_EXT.1: Password Management |
| | FIA_PSK_EXT.1: Extended: Pre-Shared Key Composition |
| | FIA_UAU.6: Re-authenticating |
| | FIA_UAU.7: Protected Authentication Feedback |
| | FIA_UAU_EXT.5: Extended: Password-based Authentication Mechanisms |
| | FIA_UIA_EXT.1: User Identification and Authentication |
| | FIA_X509_EXT.1: Extended: X509 Certificates |
| **FMT: Security management** | FMT_MOF.1: Management of Security Functions Behavior |
| | FMT_MTD.1(1): Management of TSF Data (General TSF Data) |
| | FMT_MTD.1(2): Management of TSF Data (Reading of Authentication Data) |
| | FMT_MTD.1(3): Management of TSF Data (for reading of all symmetric keys) |
| | FMT_SMF.1: Specification of management functions |
| | FMT_SMR.1: Security Management Roles |
| **FPT: Protection of the TSF** | FPT_FLS.1: Fail Secure |
| | FPT_ITT.1: Basic Internal TSF Data Transfer Protection |

| Requirement Class | Requirement Component |
|---|---|
| | FPT_RPL.1: Replay Detection |
| | FPT_STM.1: Reliable Time Stamp |
| | FPT_TST_EXT.1: Extended: TSF Testing |
| | FPT_TUD_EXT.1: Extended: Trusted Update Resource Utilization (FRU) |
| **FRU: Resource utilisation** | FRU_RSA.1: Maximum Quotas TOE Access (FTA) |
| **FTA: TOE access** | FTA_SSL.3: TSF-initiated termination |
| | FTA_SSL.4: User-initiated termination |
| | FTA_SSL_EXT.1: TSF-initiated session locking |
| | FTA_TAB.1: Default TOE Access Banners |
| | FTA_TSE.1: TOE Session Establishment Trusted Path/Channels (FTP) |
| **FTP: Trusted path/channels** | FTP_ITC.1: Inter-TSF trusted channel |
| | FTP_TRP.1: Trusted Path |

**Table 1 TOE Security Functional Components**

## 5.2.1 Security audit (FAU)

### 5.2.1.1 Audit Data Generation (FAU_GEN.1)

**FAU_GEN.1.1**

The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the not specified level of audit; and c) All administrative actions; d) [Specifically defined auditable events listed in **Table 2 Audit Events**].

| Requirement | Auditable Events | Additional Audit Record Content | Guidance Notes |
|---|---|---|---|
| FAU_GEN.1 | None | | |
| FAU_GEN.2 | None | | |
| FAU_SAR.1 | None | | |
| FAU_SAR.2 | None | | |
| FAU_SEL.1 | All modifications to the audit configuration that occur while the audit collection functions are operating. | None | The command “show audit-trail” as documented in [CLI] is used to show a log of all administrative actions. By default, only commands which change system behavior are logged. By setting the configuration parameter “audit-trail all”, all commands will be logged including commands which do not alter system behavior. |
| FAU_STG.1 | None | | |
| FAU_STG_EXT.1 | None | | |
| FAU_STG_EXT.3 | Loss of connectivity. | None | Connectivity to the audit server must be provided through an IPsec tunnel. A failure of the IPsec tunnel will indicate loss of connectivity to the audit server. See FCS_IPSEC_EXT.1 for further guidance on IPsec failure messages. |
| FCS_CKM.1(1) | Failure of the key generation activity. | None | See [SYSLOG] message ID 124865, 124866 |

| Requirement | Auditable Events | Additional Audit Record Content | Guidance Notes |
|---|---|---|---|
| FCS_CKM.1(2) | Failure of the key generation activity. | None | See [SYSLOG] message ID 103094 |
| FCS_CKM.2(1) | Failure of the key generation activity. | None | See [SYSLOG] message ID 524143 |
| FCS_CKM.2(2) | Failure of the key distribution activity, including failures related to wrapping the GTK. | Identifier(s) for intended recipients of wrapped key. | See [SYSLOG] message ID 124866 |
| FCS_CKM_EXT.4 | Failure of the key zeroization process. | Identity of subject requesting or causing zeroization, identity of object or entity being cleared. | N/A for this TOE. It is not possible for key zeroization to fail without a fatal kernel crash occurring. |
| FCS_COP.1(1) | Failure of encryption or decryption. | Cryptographic mode of operation, name/identifier of object being encrypted/decrypted. | N/A. TOE does not implement. |
| FCS_COP.1(2) | Failure of cryptographic signature. | Cryptographic mode of operation, name/identifier of object being signed/verified. | See [SYSLOG] message ID 103097, 103098, 103099, 103100 |
| FCS_COP.1(3) | Failure of hashing function. | Cryptographic mode of operation, name/identifier of object being hashed. | See [SYSLOG] message ID 103096 |
| FCS_COP.1(4) | Failure in Cryptographic Hashing for Non-Data Integrity. | Cryptographic mode of operation, name/identifier of object being hashed. | See [SYSLOG] message ID 103095 |
| FCS_COP.1(5) | Failure of WPA2 encryption or decryption. | Cryptographic mode of operation, name/identifier of object being encrypted/decrypted, non-TOE endpoint of connection (IP address). | N/A. TOE does not implement. |
| FCS_HTTPS_EXT.1 | Protocol failures. Establishment/Termination of a HTTPS session. | Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures. | See [SYSLOG] message ID 125022<br>See [SYSLOG] – Security - Warnings |
| FCS_IPSEC_EXT.1 | Protocol failures. Establishment/Termination of an IPsec SA. Negotiation “down” from an IKEv2 to IKEv1 exchange. | Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures. | See [SYSLOG] message ID 103001 through 103092<br>See [SYSLOG] message ID 103009, 103077<br>No “negotiation down” in IKE is possible, so no audit message is provided. |
| FCS_RBG_EXT.1 | Failure of the randomization process. | None | See [SYSLOG] message ID 303087, 303088, 303090. |

| Requirement | Auditable Events | Additional Audit Record Content | Guidance Notes |
|---|---|---|---|
| FCS_SSH_EXT.1 | Protocol failures.<br>Establishment/Termination of an SSH session. | Reason for failure.<br>Non-TOE endpoint of connection (IP address) for both successes and failures. | See [SYSLOG] message ID 125022<br>See [SYSLOG] – Security - Warnings |
| FCS_TLS_EXT.1 | Protocol failures.<br>Establishment/Termination of a TLS session. | Reason for failure.<br>Non-TOE endpoint of connection (IP address) for both successes and failures. | TLS is only used in the context of HTTPS. Audit messages for TLS will be the same as FCS_HTTPS_EXT.1. |
| FDP_RIP.2 | None | | |
| FIA_8021X_EXT.1 | Attempts to access to the 802.1X controlled port. | Provided client identity (IP address). | Statistics available through "show dot1x supplicant-info" and "show dot1x counters".<br><br>Note: Client identity provided by MAC address, not IP address. IP address is not applicable prior to 802.1X completion. |
| FIA_AFL.1 | The reaching of the threshold for the unsuccessful authentication attempts and the actions taken (e.g., disabling of an account) and the subsequent, if appropriate, restoration to the normal state (e.g., re-enabling of a terminal). | None | See [SYSLOG] message ID 125060 |
| FIA_PMG_EXT.1 | None | | |
| FIA_PSK_EXT.1 | None | | |
| FIA_UAU.6 | Attempts to re-authenticate. | Origin of the attempt (e.g., IP address). | Reauthentication is not treated differently than initial authentication. Audit for this activity would be identical to FIA_UIA_EXT.1. |
| FIA_UAU.7 | None | | |
| FIA_UAU_EXT.5 | All use of the authentication mechanism. | Origin of the attempt (e.g., IP address). | See [SYSLOG] – Security - Warnings |
| FIA_UIA_EXT.1 | All use of the identification and authentication mechanism. | Provided user identity, origin of the attempt (e.g., IP address). | See [SYSLOG] – Security - Warnings |
| FIA_X509_EXT.1 | Attempts to load certificates.<br>Attempts to revoke certificates. | None | Audit messages for these actions are stored in the configuration audit trail. For identification, all certificate management commands will include the keywords "crypto-local pki" with the rest of the message indicating whether a certificate |

| Requirement | Auditable Events | Additional Audit Record Content | Guidance Notes |
|---|---|---|---|
| | | | was loaded or removed. |
| FMT_MOF.1 | None | | |
| FMT_MTD.1(1) | None | | |
| FMT_MTD.1(2) | None | | |
| FMT_MTD.1(3) | None | | |
| FMT_SMF.1 | None | | |
| FMT_SMR.1 | None | | |
| FPT_FLS.1 | Failure of the TSF. | Indication that the TSF has failed with the type of failure that occurred. | See [SYSLOG] – message 303091. Also refer to FIPS Security Policy for description of audit messages. |
| FPT_ITT.1 | None | | |
| FPT_RPL.1 | Detected replay attacks. | Identity of the user that was the subject of the reply attack.<br>Identity (e.g., source IP address) of the source of the replay attack. | See [SYSLOG] – message 132093. |
| FPT_STM.1 | None | | |
| FPT_TST_EXT.1 | Execution of this set of TSF self-tests.<br>Detected integrity violations. | For integrity violations, the TSF code file that caused the integrity violation. | See “Self Test” section of [FIPS] for details. |
| FPT_TUD_EXT.1 | Initiation of the update.<br>Any failure to verify the integrity of the update. | None | The audit trail will indicate when a new software image has been copied to the TOE through use of the “copy” command. A complete reboot is required to make an update actually take effect. |
| FRU_RSA.1 | Maximum quota being exceeded. | Resource identifier. | See [SYSLOG] message ID 124008. The message reason will indicate “Monitor/police CP attacks”. |
| FTA_SSL.3 | The termination of a remote session by the session locking mechanism. | None | See [SYSLOG] – Security - Warnings |
| FTA_SSL.4 | Terminating a session by quitting or logging off. | None | See [SYSLOG] – Security - Warnings |
| FTA_SSL_EXT.1 | Locking of an interactive session by the session locking mechanism.<br>Any attempts at unlocking of an interactive session. | None | N/A for this TOE. Interactive sessions are only terminated, not locked. |
| FTA_TAB.1 | None | | |
| FTA_TSE.1 | Denial of a session establishment due to the session establishment | Reason for denial, origin of establishment attempt. | See [SYSLOG] – message ID 522039, 124006. |

| Requirement | Auditable Events | Additional Audit Record Content | Guidance Notes |
|---|---|---|---|
| | mechanism. | | |
| FTP_ITC.1 | All attempts to establish a trusted channel.<br>Detection of modification of channel data. | Identification of the initiator and target of channel. | The Inter-TSF trusted channel is IPsec. Audit messages will be the same as for FCS_IPSEC_EXT.1. |
| FTP_TRP.1 | All attempts to establish a remote administrative session.<br>Detection of modification of session data. | Identification of the initiating IT entity (e.g., IP address). | Depending on whether the remote administrator is using HTTPS or SSH, the audit messages will be the same as FCS_SSH_EXT.1 or FCS_HTTPS_EXT.1. Audit message 125022 includes the identification of the claimed user identity. |

**Table 2 Audit Events**

**Assurance Activity:**

The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Table 9.

The evaluator shall in particular ensure that the operational guidance is clear in relation to the contents for failed cryptographic events. In Table 9, information detailing the cryptographic mode of operation and a name or identifier for the object being encrypted is required. The evaluator shall ensure that name or identifier is sufficient to allow an administrator reviewing the audit log to determine the context of the cryptographic operation (for example, performed during a key negotiation exchange, performed when encrypting data for transit) as well as the non-TOE endpoint of the connection for cryptographic failures relating to communications with other IT systems.

The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. The TOE may contain functionality that is not evaluated in the context of this PP because the functionality is not specified in an SFR. This functionality may have administrative aspects that are described in the operational guidance. Since such administrative actions will not be performed in an evaluated configuration of the TOE, the evaluator shall examine the operational guidance and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP, which thus form the set of 'all administrative actions'. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records in accordance with the assurance activities associated with the functional requirements in this PP. Additionally, the evaluator shall test that each administrative action applicable in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

Note that the testing here can be accomplished in conjunction with the testing of the security

mechanisms directly. For example, testing to ensure the TOE can detect replay attempts will more than likely be done to demonstrate that requirement FPT_RPL.1 is satisfied. Another example is that testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

**FAU_GEN.1.2**

The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [information specified in column three of **Table 2 Audit Events**].

**Assurance Activity:**

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

#### 5.2.1.2 User Audit Association (FAU_GEN.2)

**FAU_GEN.2.1**

For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

**Component Assurance Activity:**

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

#### 5.2.1.3 Audit Review (FAU_SAR.1)

**FAU_SAR.1.1**

The TSF shall provide Authorized Administrators with the capability to read all audit data from the audit records.

**FAU_SAR.1.2**

Refinement: The TSF shall provide the audit records in a manner suitable for the ~~user~~ Authorized Administrators to interpret the information.

#### 5.2.1.4 Restricted Audit Review (FAU_SAR.2)

**FAU_SAR.2.1**

Refinement: The TSF shall prohibit all users read access to the audit records in the audit trail, except Authorized Administrators.

#### 5.2.1.5 Selective Audit (FAU_SEL.1)

**FAU_SEL.1.1**

The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes: a) event type; b) success of auditable security events; c) failure of auditable security events; and d) [**device interface and wireless client identity**].

**Assurance Activity:**

The evaluator shall review the administrative guidance to ensure that the guidance itemizes all event types, as well as describes all attributes that are to be selectable in accordance with the requirement, to include those attributes listed in the assignment. The administrative guidance shall also contain instructions on how to set the pre-selection, as well as explain the syntax (if present) for multi-value pre-selection. The administrative guidance shall also identify those audit records that are always recorded, regardless of the selection criteria currently being enforced.

The evaluator shall also perform the following tests:

Test 1: For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.

Test 2 [conditional]: If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.

#### 5.2.1.6 Protected Audit Trail Storage (Local Storage) (FAU_STG.1)

**FAU_STG.1.1**

Refinement: The TSF shall protect [**3*(3*31768 bytes)**] locally stored audit records in the audit trail from unauthorized deletion.

**FAU_STG.1.2**

The TSF shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.

**Component Assurance Activity:**

The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access. The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and 'cleared' periodically by sending the data to the audit server.

#### 5.2.1.7 External Audit Trail Storage (FAU_STG_EXT.1)

**FAU_STG_EXT.1.1**

The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel implementing the [***IPsec***] protocol.

**Component Assurance Activity:**

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. The evaluator shall perform the following test for this requirement:

Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.

#### 5.2.1.8 Action in Case of Loss of Audit Server Connectivity (FAU_STG_EXT.3)

**FAU_STG_EXT.3.1**

The TSF shall [**generate a local log message indicating failure of an IPsec tunnel**] if the link to the external IT entity collecting the audit data generated by the TOE is not available.

**Component Assurance Activity:**

The evaluator shall examine the administrative guidance to ensure it instructs the administrator how to establish communication with the audit server. The guidance must instruct how this channel is established in a secure manner (e.g., IPsec, TLS). The evaluator checks the administrative guidance to determine what action(s) is taken if the link between the TOE and audit server is broken. This could be due to network connectivity being lost, or the secure protocol link being terminated.

The evaluator shall examine the operational guidance to determine any activities that must take place after connectivity is restored to ensure that local audit events captured during the period of loss are synchronized with the audit trail on the audit server, and informs the administrator of any limitations on the data that are able to be sent (for instance, if the duration of the outage is significant, the local store may not contain all of the records that where generated during this period).

The evaluator shall perform the following test for this requirement:

Test 1: The evaluator shall test the administrative guidance by establishing a link to the audit server. Note that this will need to be done in order to perform the assurance activities prescribed under FAU_GEN.1. The evaluator shall disrupt the communication link (e.g., unplug the network cable, terminate the protocol link, shutdown the audit server) to determine that the action(s) described in the administrative guide appropriately take place.

## 5.2.2 Cryptographic support (FCS)

### 5.2.2.1 Cryptographic Key Generation (Symmetric Keys for WPA2 Connections) (FCS_CKM.1(1))

**FCS_CKM.1.1(1)**

Refinement: The TSF shall derive symmetric cryptographic keys in accordance with a specified cryptographic key derivation algorithm [PRF-384] with specified cryptographic key size [128 bits] using a Random Bit Generator as specified in FCS_RBG_EXT.1 and that meet the following: [802.11-2007].

**Component Assurance Activity:**

The cryptographic primitives will be verified through assurance activities specified later in this PP. The evaluator shall verify that the TSS describes how the primitives defined and implemented by this PP are used by the TOE in establishing and maintaining secure connectivity to the wireless clients. The TSS shall also provide a description of the developer's method(s) of assuring that their implementation conforms to the cryptographic standards; this includes not only testing done by the developing organization, but also any third-party testing that is performed. The evaluator shall ensure that the description of the testing methodology is of sufficient detail to determine the extent to which the details of the protocol specifics are tested.

### 5.2.2.2 Cryptographic Key Generation (Asymmetric Keys) (FCS_CKM.1(2))

**FCS_CKM.1.1(2)**

Refinement: The TSF shall generate asymmetric cryptographic keys used for key establishment in accordance with [

- ***NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for elliptic curve-based key establishment schemes and implementing 'NIST curves' P-256, P-384 and [no other curves] (as defined in FIPS PUB 186-3, 'Digital Signature Standard');***
- ***NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography' for RSA-based key establishment schemes***]

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

**Component Assurance Activity:**

The evaluator shall use the key pair generation portions of 'The FIPS 186-3 Digital Signature Algorithm Validation System (DSA2VS)', 'The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)', and 'The RSA Validation System (RSA2VS)' as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

In order to show that the TSF implementation complies with 800-56A and/or 800-56B, depending on the selections made, the evaluator shall ensure that the TSS contains the following information:

The TSS shall list all sections of the appropriate 800-56 standard(s) to which the TOE complies.

For each applicable section listed in the TSS, for all statements that are not 'shall' (that is, 'shall not', 'should', and 'should not'), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as 'shall not' or 'should not' in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;

For each applicable section of 800-56A and 800-56B (as selected), any omission of functionality related to 'shall' or 'should' statements shall be described;

Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described.

#### 5.2.2.3 Cryptographic Key Distribution (PMK) (FCS_CKM.2(1))

**FCS_CKM.2.1(1)**

Refinement: The TSF shall distribute the 802.11 Pairwise Master Key in accordance with a specified cryptographic key distribution method: [receive from 802.1X Authorization Server] that meets the following: [802.11-2007] and does not expose the cryptographic keys.

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that it describes how the PMK is transferred (that is, through what EAP attribute) to the TSF.

The evaluator shall perform the following test:

Test 1: The evaluator shall establish a session between the TOE and a RADIUS server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the RADIUS server and the TOE during a successful attempt to connect a wireless client to the TOE to determine that the PMK is not exposed.

#### 5.2.2.4 Cryptographic Key Distribution (GTK) (FCS_CKM.2(2))

**FCS_CKM.2.1(2)**

Refinement: The TSF shall distribute Group Temporal Key in accordance with a specified cryptographic key distribution method: [AES Key Wrap in an EAPOL-Key frame] that meets the following: [RFC 3394 for AES Key Wrap, 802.11-2007 for the packet format and timing considerations] and does not expose the cryptographic keys.

**Component Assurance Activity:**

The evaluator shall check the TSS to ensure that it describes how the GTK is wrapped prior to be distributed using the AES implementation specified in this PP, and also how the GTKs are

distributed when multiple clients connect to the TOE. The evaluator shall also perform the following test:

Test 1: The evaluator shall successfully connect multiple clients to the TOE. As the clients are connected, the evaluator shall observe that the GTK is not transmitted in the clear between the client and the TOE.

Test 2: The evaluator shall cause a broadcast message to be sent to all clients connected to the TOE. The evaluator shall ensure the message is encrypted and cannot be read.

Test 3: The evaluator shall create at least two multicast groups among a subset of clients connected to the TOE, each consisting of at least two clients but less than all of the clients connected to the TOE. Some (but not all) of the clients shall be in both groups. The evaluator shall ensure that GTKs established are sent to the participating clients and cannot be determine from the traffic flowing between the clients and the TOE.

Test 4: The evaluator shall cause a multicast message to be sent to the clients in each multicast group connected to the TOE. The evaluator shall ensure each message is encrypted and cannot be read.

#### 5.2.2.5 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

**FCS_CKM_EXT.4.1**

The TSF shall zeroize all plaintext secret and private cryptographic keys and cryptographic security parameters when no longer required.

**Component Assurance Activity:**

The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and critical security parameters used to generate keys; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the zeroization procedure in terms of the type of the memory or storage in which the data are stored (for example, 'secret keys stored on flash are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write').

#### 5.2.2.6 Cryptographic Operation (Data Encryption/Decryption) (FCS_COP.1(1))

**FCS_COP.1.1(1)**

Refinement: The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES operating in [**CCM or GCM mode**]] and cryptographic key sizes 128-bits, 256-bits, and [***192 bits***] that meets the following: FIPS PUB 197, 'Advanced Encryption Standard (AES)' **and** [***NIST SP 800-38A, NIST SP 800-38C, NIST SP 800-38D***].

**Component Assurance Activity:**

The evaluator shall use tests appropriate to the modes selected in the above requirement from 'The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)', 'The XTS-AES Validation System (XTSVS)', 'The CMAC Validation System (CMACVS)', 'The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)', and 'The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)' (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

#### 5.2.2.7 Cryptographic Operation (Cryptographic Signature) (FCS_COP.1(2))

**FCS_COP.1.1(2)**

Refinement: The TSF shall perform cryptographic signature services in accordance with a [

***(2) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater, or***

***(3) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater***]

that meets the following:

Case: RSA Digital Signature Algorithm

[***FIPS PUB 186-3, 'Digital Signature Standard'***]

Case: Elliptic Curve Digital Signature Algorithm

[***FIPS PUB 186-3, 'Digital Signature Standard'***]

The TSF shall implement 'NIST curves' P-256, P-384 and [***no other curves***] (as defined in FIPS PUB 186-3, 'Digital Signature Standard').

**Component Assurance Activity:**

The evaluator shall use the signature generation and signature verification portions of 'The Digital Signature Algorithm Validation System' (DSA2VS), 'The Elliptic Curve Digital Signature Algorithm Validation System' (ECDSA2VS), and 'The RSA Validation System' (RSA2VS) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e. FIPS PUB 186-3). This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

#### 5.2.2.8 Cryptographic Operation (Cryptographic Hashing) (FCS_COP.1(3))

**FCS_COP.1.1(3)**

Refinement: The TSF shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384***] and message digest sizes [***160, 256, 384***] bits that meet the following: FIPS Pub 180-3, 'Secure Hash Standard.'

**Component Assurance Activity:**

The evaluator shall use 'The Secure Hash Algorithm Validation System (SHAVS)' as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

#### 5.2.2.9 Cryptographic Operation (Keyed-Hash Message Authentication) (FCS_COP.1(4))

**FCS_COP.1.1(4)**

Refinement: The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC- [***SHA-1, SHA-256, SHA-384, SHA-1-96***], key size [**128,256**], and message digest size of [***160, 256, 384***] bits that meet the following: FIPS PUB 198-1, 'The Keyed-Hash Message Authentication Code', and FIPS PUB 180-3, 'Secure Hash Standard'.

**Component Assurance Activity:**

The evaluator shall use 'The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS)' as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

#### 5.2.2.10 Cryptographic Operation (WPA2 Data Encryption/Decryption) (FCS_COP.1(5))

**FCS_COP.1.1(5)**

Refinement: The TSF shall perform encryption and decryption in accordance with the specified cryptographic algorithm AES CCMP and cryptographic key size of 128 bits that meet the following: FIPS PUB 197, NIST SP 800-38C and IEEE 802.11-2007.

**Component Assurance Activity:**

The evaluator shall use tests from "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)" as a guide in testing the requirement above. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

Additionally, the evaluator shall use tests from the IEEE 802.11-02/362r6 document "Proposed Test vectors for IEEE 802.11 TGi", dated September 10, 2002, Section 2.1 AES-CCMP Encapsulation Example and Section 2.2 Additional AES CCMP Test Vectors to further verify the IEEE 802.11-2007 implementation of AES-CCMP.

#### 5.2.2.11 Extended: HTTP Security (HTTPS) (FCS_HTTPS_EXT.1)

**FCS_HTTPS_EXT.1.1**

The TSF shall implement the HTTPS protocol that complies with RFC 2818.

**Assurance Activity:**

In order to show that the TSF implements the RFCs correctly, the evaluator shall ensure that the TSS contains the following information:

For each section of each applicable RFC listed for the FCS_HTTPS_EXT.1 elements, for all statements that are not 'MUST' (for example, 'MAY', 'SHOULD', 'SHOULD NOT', etc.), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as 'SHOULD NOT' or 'MUST NOT' in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;

For each section of each RFC, any omission of functionality related to 'MUST' or 'SHOULD' statements shall be described;

Any TOE-specific extensions, processing that is not included in the standard, or alternative implementations allowed by the standard that may impact the security requirements the TOE is to enforce shall be described.

**FCS_HTTPS_EXT.1.2**

The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

**Assurance Activity:**

The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. administrator authentication which may be done at a different level of the processing stack. Testing for this activity is done as part of the TLS testing; this may result in additional testing if the TLS tests are done at the TLS protocol level.

#### 5.2.2.12 Extended: Internet Protocol Security (IPsec) Communications (FCS_IPSEC_EXT.1)

**FCS_IPSEC_EXT.1.1**

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602), [***AES-GCM-128, AES-GCM-256 as specified in RFC 4106***], and using [***IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, and [RFC 4868 for hash functions]; IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and [RFC 4868 for hash functions]***] for connections to the Authentication Server and [***[audit and NTP servers]***].

**Assurance Activity:**

In order to show that the TSF implements the RFCs correctly, the evaluator shall ensure that the TSS contains the following information:

For each section of each applicable RFC listed for the FCS_IPSEC_EXT.1 elements, for

all statements that are not 'MUST' (for example, 'MAY', 'SHOULD', 'SHOULD NOT', etc.), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as 'SHOULD NOT' or 'MUST NOT' in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;

For each section of each RFC, any omission of functionality related to 'MUST' or 'SHOULD' statements shall be described;

Any TOE-specific extensions, processing that is not included in the standard, or alternative implementations allowed by the standard that may impact the security requirements the TOE is to enforce shall be described.

The evaluator shall ensure the TSS identifies all servers/services that require or allow IPsec connections. The evaluators shall also ensure that when performing testing and analysis activities, the activities apply to all servers identified. The evaluators shall ensure that at least one instance of every type of server is used in at least one test during the testing activities to provide assurance that the identified communications can take place. The evaluators shall also ensure that the configuration information (including product and version numbers) for the non-TOE endpoints of these connections is recorded in the test report.

The evaluator shall also perform the following test for TOEs that implement IKEv2:

Test 1 [conditional]: The evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 4306, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

**FCS_IPSEC_EXT.1.2**

The TSF shall ensure that only ESP confidentiality and integrity security service is used.

**Assurance Activity:**

The evaluator shall examine the TSS to verify that it describes how the 'confidentiality only' ESP security service is disabled. The evaluator shall also examine the operational guidance to determine that it describes any configuration necessary to ensure negotiation of 'confidentiality only' security service for ESP is disabled, and that an advisory is present indicating that tunnel mode is the preferred ESP mode since it protects the entire packet.

Test 1: The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using ESP using the 'confidentiality only' security service. This attempt should fail. The evaluator shall then establish a connection using ESP using the confidentiality and integrity security service.

**FCS_IPSEC_EXT.1.3**

The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

**Assurance Activity:**

The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. If this requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance. The evaluator shall also perform the following tests:

Test 1: The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported.

**FCS_IPSEC_EXT.1.4**

The TSF shall ensure that [***IKEv1 SA lifetimes are able to be limited by number of packets and time: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs; IKEv2 SA lifetimes can be configured by an administrator based on number of packets or length of time***].

**Assurance Activity:**

If IKEv1 requirements are selected, the evaluator checks to ensure that the TSS describes how lifetimes for IKEv1 SAs (both Phase 1 and Phase 2) are established. If they are configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance. For IKEv2 requirements, the evaluator verifies that the values can be configured and that the instructions for doing so are located in the operational guidance. The evaluator also performs the following tests, depending on whether IKEv1, IKEv2, or both are configured:

Test 1 (IKEv1): The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.

Test 2 (IKEv1): The evaluator shall perform a test similar to Test 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.

Test 3 (IKEv1 and v2): The evaluator shall configure a maximum lifetime in terms of the # of packets allowed; this may be a hard-coded value for IKEv1, otherwise, the evaluator follows the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets through this SA is exceeded, the connection is closed.

Test 4 (IKEv2): The evaluator shall configure a time-based maximum lifetime for an SA, and then establish the SA. The evaluator shall observe that this SA is closed or renegotiated in the established time.

**FCS_IPSEC_EXT.1.5**

The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ('x' in gx mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [**112, 160, 256, and 384**] bits.

**Assurance Activity:**

The evaluator shall check to ensure that, for each DH group supported by the TSF, the TSS describes the process for generating 'x'(as defined in FCS_IPSEC_EXT.1.5) and each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of 'x' and the nonces meet the stipulations in the requirement.

**FCS_IPSEC_EXT.1.6**

The TSF shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^[**80, 128, and 192**].

**Assurance Activity:**

See the Assurance Activity for FCS_IPSEC_EXT.1.5.

**FCS_IPSEC_EXT.1.7**

The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP) and***, [19 (256-bit Random ECP), 20 (384-bit Random ECP), [DH MODP Group 2 (1024-bit group)]]***

**Assurance Activity:**

The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer. The evaluator shall also perform the following test:

Test 1: For each supported DH group, the evaluator shall test to ensure that all IKE protocols can be successfully completed using that particular DH group.

**FCS_IPSEC_EXT.1.8**

The TSF shall ensure that all IKE protocols implement peer authentication using Pre-shared Keys and [**rDSA, ECDSA**] that use X.509v3 certificates that conform to RFC 4945.

**Assurance Activity:**

The evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established for a TOE. The description in the TSS and the operational guidance shall also indicate how pre-shared key establishment is accomplished for both TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key. The evaluator shall also perform the following test:

Test 1: The evaluator shall generate a pre-shared key and use it, as indicated in the operational guidance, to establish an IPsec connection between two peers. If the TOE supports generation of the pre-shared key, the evaluator shall ensure that establishment of the key is carried out for an instance of the TOE generating the key as well as an instance of the TOE merely taking in and using the key.

The evaluator shall check that the TSS contains a description of the IKE peer authentication process used by the TOE, and that this description covers the use of the algorithm or algorithms specified in the selection. As part of the assurance activity for FCS_IPSEC_EXT.1.1, required and optional elements of RFC 4945 shall be documented. The evaluator shall also perform the following tests:

Test 1: For each supported algorithm, the evaluator shall test that peer authentication using that algorithm can be successfully achieved.

Test 2: For each supported identification payload (from RFC 4945), the evaluator shall test that peer authentication can be successfully achieved.

Test 3: The evaluator shall devise a test that demonstrates that a corrupt or invalid certification path for a certificate will be detected during IKE peer authentication and will result in a connection not being established.

Test 4: The evaluator shall devise a test that demonstrates that a certificate that has been revoked through a CRL will be detected during IKE peer authentication and will result in a connection not being established.

**FCS_IPSEC_EXT.1.9**

The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv1 Phase 1, IKEv2 IKE_SA***] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv1 Phase 2, IKEv2 CHILD_SA***] connection.

**Assurance Activity:**

The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The

TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation. The evaluator shall also perform the following tests:

Test 1: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.

Test 2: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.

#### 5.2.2.13 Extended: Cryptographic Operation: Random Bit Generation (FCS_RBG_EXT.1)

**FCS_RBG_EXT.1.1**

The TSF shall perform all random bit generation (RBG) services in accordance with [***NIST Special Publication 800-90*** using [***CTR_DRBG(any)***]] seeded by an entropy source that accumulates entropy from at least one independent TSF-hardware-based noise sources

**FCS_RBG_EXT.1.2**

The deterministic RBG shall be seeded with a minimum of [***256 bits***] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

**Component Assurance Activity:**

The evaluator shall review the TSS section to determine the version number of the product containing the RBG(s) used in the TOE. The evaluator shall also confirm that the TSS describes the hardware-based noise source from which entropy is gathered, and further confirm that this noise source is located on the USB Flash Drive. The evaluator will further verify that all of the underlying functions and parameters used in the RBG are listed in the TSS.

The evaluator shall verify that the TSS contains a description of the RBG model, including the method for obtaining entropy input, as well as identifying the entropy source(s) used, how entropy is produced/gathered from each source, and how much entropy is produced by each entropy source. The evaluator shall also ensure that the TSS describes the entropy source health tests, a rationale for why the health tests are sufficient to determine the health of the entropy sources, and known modes of entropy source failure. Finally, the evaluator shall ensure that the TSS contains a description of the RBG outputs in terms of the independence of the output and variance with time and/or environmental conditions.

Regardless of the standard to which the RBG is claiming conformance, the evaluator perform the following test:

Test 1: The evaluator shall determine an entropy estimate for each entropy source by using the Entropy Source Test Suite. The evaluator shall ensure that the TSS includes an entropy estimate that is the minimum of all results obtained from all entropy sources.

The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.

**Implementations Conforming to FIPS 140-2, Annex C**

The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluators shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.

The evaluators shall perform a Variable Seed Test. The evaluators shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluators ensure that the values returned by the TSF match the expected values.

The evaluators shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluators then invoke the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3. The evaluators ensure that the 10,000th value produced matches the expected value.

#### 5.2.2.14 Extended: Secure Shell (SSH) (FCS_SSH_EXT.1)

**FCS_SSH_EXT.1.1**

The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254.

**Assurance Activity:**

In order to show that the TSF implements the RFCs correctly, the evaluator shall ensure that the TSS contains the following information:

For each section of each applicable RFC listed for the FCS_SSH_EXT.1 elements, for all statements that are not 'MUST' (for example, 'MAY', 'SHOULD', 'SHOULD NOT', etc.), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as 'SHOULD NOT' or 'MUST NOT' in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;

For each section of each RFC, any omission of functionality related to 'MUST' or 'SHOULD' statements shall be described;

Any TOE-specific extensions, processing that is not included in the standard, or alternative implementations allowed by the standard that may impact the security requirements the TOE is to enforce shall be described.

**FCS_SSH_EXT.1.2**

The TSF shall ensure that the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

**Assurance Activity:**

The evaluator shall examine the TSS to ensure that it specifies that the TOE rekeys an SSH connection before more than 2^28 packets have been sent with a given key. If this effect is achieved by configuration of the TOE, then the evaluator shall examine the operational guidance to ensure that it contains instructions on setting the appropriate values.

**FCS_SSH_EXT.1.3**

The TSF shall ensure that the SSH protocol implements a timeout period for authentication as defined in RFC 4252 of [**30 seconds**], and provide a limit to the number of failed authentication attempts a client may perform in a single session to [**3**] attempts.

**Assurance Activity:**

The evaluator shall check to ensure that the TSS specifies the timeout period and the method for dropping a session connection after the number of failed authentication attempts specified in the requirement. If these values are configurable and may be specified by the administrator, the

evaluator shall check the operational guidance to ensure that it contains instructions for configuring these values. The evaluator shall also perform the following tests:

Test 1: The evaluator shall demonstrate that taking longer than the timeout period to authenticate to the TOE results in a disconnection of the current session and requires that the evaluator initiate a new session to attempt to connect. If the timeout period is configurable, the evaluator shall ensure that the operational guidance is followed to implement at least two different periods in order to ensure that the mechanism works as specified.

Test 2: The evaluator shall demonstrate that performing a number of failed SSH authentication attempts equal to the value specified in the requirement results in a disconnection of the current session and requires that the evaluator initiate a new session to attempt to connect. If this number is configurable, the evaluator shall ensure that the operational guidance is followed to implement at least two different limits (e.g., 3 attempts and 5 attempts) in order to ensure that the mechanism works as specified.

**FCS_SSH_EXT.1.4**

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

**Assurance Activity:**

The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.7, and ensure that password-based authentication methods are also allowed. The evaluator shall also perform the following tests:

Test 1: The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.

Test 2: Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

**FCS_SSH_EXT.1.5**

The TSF shall ensure that, as described in RFC 4253, packets greater than [**32,768**] bytes in an SSH transport connection are dropped.

**Assurance Activity:**

The evaluator shall check that the TSS describes how 'large packets' in terms of RFC 4253 are detected and handled. The evaluator shall also perform the following test:

Test 1: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

**FCS_SSH_EXT.1.6**

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256-CBC, [**no other encryption algorithms**].

**Assurance Activity:**

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE

may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

Test 1: The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of a protocol to satisfy the intent of the test.

**FCS_SSH_EXT.1.7**

The TSF shall ensure that the SSH transport implementation uses SSH_RSA and [***no other public key algorithms***] as its public key algorithm(s).

**Assurance Activity:**

The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

**FCS_SSH_EXT.1.8**

The TSF shall ensure that the data integrity algorithm used in the SSH transport connection is hmac-sha1 and [***hmac-sha1-96***].

**Assurance Activity:**

The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the 'none' MAC algorithm is not allowed).

**FCS_SSH_EXT.1.9**

The TSF shall ensure that diffie-hellman-group14-sha1 is the only allowed key exchange method used for the SSH protocol.

**Assurance Activity:**

The evaluator shall ensure that operational guidance contains configuration information that will allow an authorized administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14. If this capability is 'hard-coded' into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol. The evaluator shall also perform the following test:

Test 1: The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. The evaluator shall then attempt to perform a diffie-hellman-group14-sha1 key exchange, and observe that the attempt succeeds.

### 5.2.2.15 Extended: Transport Layer Security (TLS) (FCS_TLS_EXT.1)

**FCS_TLS_EXT.1.1**

The TSF shall implement one or more of the following protocols [TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), and TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

Mandatory Ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA,
TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
TLS_DHE_RSA_WITH_AES_256_CBC_SHA;

Optional Ciphersuites:

[
***TLS_RSA_WITH_AES_128_CBC_SHA256,***
***TLS_RSA_WITH_AES_256_CBC_ SHA256,***
***TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256,***
***TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256,***
***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,***

***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,***
***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,***
***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384*** ].

**Component Assurance Activity:**

In order to show that the TSF implements the RFCs correctly, the evaluator shall ensure that the TSS contains the following information:

For each section of each applicable RFC listed for the FCS_TLS_EXT.1 elements, for all statements that are not 'MUST' (for example, 'MAY', 'SHOULD', 'SHOULD NOT', etc.), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as 'SHOULD NOT' or 'MUST NOT' in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;

For each section of each RFC, any omission of functionality related to "MUST" or “SHOULD” statements shall be described;

Any TOE-specific extensions, processing that is not included in the standard, or alternative implementations allowed by the standard that may impact the security requirements the TOE is to enforce shall be described.

The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe (on the wire) the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

## 5.2.3 User data protection (FDP)

### 5.2.3.1 Full Resident Information Protection (FDP_RIP.2)

**FDP_RIP.2.1**

The TSF shall enforce that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

**Component Assurance Activity:**

'Resources' in the context of this requirement are network packets being sent through (as opposed to 'to', as is the case when an administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

## 5.2.4 Identification and authentication (FIA)

### 5.2.4.1 Extended: 802.1X Port Access Entity (Authenticator) Authentication (FIA_8021X_EXT.1)

**FIA_8021X_EXT.1.1**

The TSF shall conform to IEEE Standard 802.1X for a Port Access Entity (PAE) in the 'Authenticator' role.

**FIA_8021X_EXT.1.2**

The TSF shall support communications to a RADIUS authentication server conforming to RFCs 2865 and 3579.

**FIA_8021X_EXT.1.3**

The TSF shall ensure that no access to its 802.1X controlled port is given to the wireless client prior to successful completion of this authentication exchange.

**Component Assurance Activity:**

In order to show that the TSF implements the 802.1X-2010 standard correctly, the evaluator shall ensure that the TSS contains the following information:

- the sections (clauses) of the standard that the TOE implements;
- For each identified section, any options allowed by the standards are specified; and
- For each identified section, any non-conformance is identified and described, including a justification for the non-conformance.

Because the connection to the RADIUS server will be contained in an IPsec tunnel (FCS_IPSEC_EXT.1), the security mechanisms detailed in the RFCs identified in the requirement are not relied on to provide protection for these communications. Consequently, no extensive analysis of the RFCs is required. However, the evaluator shall ensure that the TSS describes the measures (documentation, testing) that are taken by the product developer to ensure that the TOE conforms to the RFCs listed in this requirement.

The evaluator shall also perform the following tests:

- Test 1: The evaluator shall demonstrate that a wireless client has no access to the test network. After successfully authenticating with a RADIUS server through the TOE, the evaluator shall demonstrate that the wireless client does have access to the test network.
- Test 2: The evaluator shall demonstrate that a wireless client has no access to the test network. The evaluator shall attempt to authenticate using an invalid client certificate, such that the EAP-TLS negotiation fails. This should result in the wireless client still being unable to access the test network.
- Test 3: The evaluator shall demonstrate that a wireless client has no access to the test network. The evaluator shall attempt to authenticate using an invalid RADIUS certificate, such that the EAP-TLS negotiation fails. This should result in the wireless client still being unable to access the test network.

It should be noted that tests 2 and 3 above are not tests that "EAP-TLS works", although that's a by-product of the test. The test is actually that a failed authentication (under two failure modes) results in denial of access to the network, which is the 3rd element of this component.

### 5.2.4.2 Authentication Failure Handling (FIA_AFL.1)

**FIA_AFL.1.1**

Refinement: The TSF shall detect when an Authorized Administrator configurable positive integer of successive unsuccessful authentication attempts occur related to administrators attempting to authenticate remotely.

**FIA_AFL.1.2**

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [***prevent the offending remote administrator from successfully authenticating until an Authorized Administrator defined time period has elapsed***].

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability. The evaluator shall also examine the operational guidance to ensure that instructions for configuring the number of successive unsuccessful authentication attempts (1.1) and time period (1.2, if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each 'action' specified (if that option is chosen). If different actions or mechanisms are implemented depending on the authentication method (e.g., TSL vs. SSH), all must be described.

The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g., TLS, SSH):

Test 1 [conditional on first selection item]: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE. The evaluator shall test that once the limit is reached, attempts with valid credentials are not successful. For each action specified by the requirement, the evaluator shall show that following the operational guidance and performing each action to allow the remote administrator access are successful.

Test 2 [conditional on second selection item]: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE and a time period after which valid logins will be allowed for a remote administrator. After exceeding the specified number of invalid login attempts and showing that valid login is not possible, the evaluator shall show that waiting for the interval defined by the time period before another access attempt will result in the ability for the remote administrator to successfully log on using valid credentials.

#### 5.2.4.3 Password Management (FIA_PMG_EXT.1)

**FIA_PMG_EXT.1.1**

The TSF shall provide the following password management capabilities for administrative passwords:

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters (that include: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')');
2. Minimum password length shall be settable by the Authorized Administrator, and support passwords of 8 characters or greater;
3. Passwords composition rules specifying the types and number of required characters that comprise the password shall be settable by the Administrator.
4. Passwords shall have a maximum lifetime, configurable by the Authorized Administrator.
5. New passwords must contain a minimum of 4 character changes from the previous password.

**Component Assurance Activity:**

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length; the formulation and specification of password composition rules and how to configure these for the TOE; and how to configure the maximum lifetime for a password. The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case.

Test 1: The evaluator shall configure the TOE with different password composition rules, as

specified in the requirement. The evaluator shall then, for each set of rules, compose passwords that both meet the requirements, and fail to meet the requirements, in some way. For each password, the evaluator shall verify that the composition rules are enforced. While the evaluator is not required (nor is it feasible) to test all possible composition rules, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

Test 2: The evaluator shall ensure that the operational guidance contains instructions on setting the maximum password lifetime. The evaluator shall then configure this lifetime to several values, and ensure that it is enforced for each of those values.

Test 3: The evaluator shall test that a minimum of 4 character changes from previous passwords is enforced. This shall be done for more than one password.

#### 5.2.4.4 Extended: Pre-Shared Key Composition (FIA_PSK_EXT.1)

**FIA_PSK_EXT.1.1**

The TSF shall be able to use pre-shared keys for IPsec and [***WPA2***].

**FIA_PSK_EXT.1.2**

The TSF shall be able to accept text-based pre-shared keys that: are 22 characters and [***maximum 64 characters]*** composed of any combination of upper and lower case letters, numbers, and special characters (that include: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')').

**FIA_PSK_EXT.1.3**

The TSF shall condition the text-based pre-shared keys by using [PBKDF2].

**FIA_PSK_EXT.1.4**

The TSF shall be able to [***accept***] bit-based pre-shared keys.

**Component Assurance Activity:**

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.

The evaluator shall examine the TSS to ensure that it identifies all protocols that allow both text-based and bit-based pre-shared keys, and states that text-based pre-shared keys of 22 characters are supported. For each protocol identified by the requirement, the evaluator shall confirm that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to the bit string used by the protocol, and that this conditioning is consistent with the last selection in the FIA_PSK_EXT.1.3 requirement.

The evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be performed with a single test case.

Test 1: The evaluator shall compose a pre-shared key of 22 characters that contains a combination of the allowed characters in accordance with the operational guidance, and demonstrates that a successful protocol negotiation can be performed with the key.

Test 2 [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall

repeat Test 1 using the minimum length; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE.

Test 3 [conditional]: If the TOE does not generate bit-based pre-shared keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

Test 4 [conditional]: If the TOE does generate bit-based pre-shared keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

#### 5.2.4.5 Re-authenticating (FIA_UAU.6)

**FIA_UAU.6.1**

The TSF shall re-authenticate the administrative user under the conditions: when the user changes their password, [***no other conditions***].

Application Note: Only administrators with "root" privilege level can change passwords. Users without "root" privilege cannot change any password, even their own.

**Component Assurance Activity:**

The evaluator shall perform the following test for each of the conditions specified in the requirement:

Test 1: The evaluator shall attempt to change their password as directed by the operational guidance. While making this attempt, the evaluator shall verify that re-authentication is required.

#### 5.2.4.6 Protected Authentication Feedback (FIA_UAU.7)

**FIA_UAU.7.1**

The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

**Component Assurance Activity:**

The evaluator shall perform the following test for each method of local login allowed:

Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

#### 5.2.4.7 Extended: Password-based Authentication Mechanisms (FIA_UAU_EXT.5)

**FIA_UAU_EXT.5.1**

The TSF shall provide a local password-based authentication mechanism, [ ***[LDAP, RADIUS, and TACACS+-based authentication]***] to perform administrative user authentication.

**FIA_UAU_EXT.5.2**

The TSF shall ensure that administrative users with expired passwords are [***locked out until their password is reset by an administrator***].

**Component Assurance Activity:**

Assurance activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

#### 5.2.4.8 User Identification and Authentication (FIA_UIA_EXT.1)

**FIA_UIA_EXT.1.1**

The TSF shall allow responses to the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process: Display the warning banner in accordance with FTA_TAB.1; [**no other services.**]

**FIA_UIA_EXT.1.2**

The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a 'successful logon'. The evaluator shall examine the operational guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting the allowed services.

The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

Test 1: The evaluator shall use the operational guidance to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

Test 2: The evaluator shall configure the services allowed (if any) according to the operational guidance, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.

Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

#### 5.2.4.9 Extended: X509 Certificates (FIA_X509_EXT.1)

**FIA_X509_EXT.1.1**

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec connections and [**no other protocols**].

**FIA_X509_EXT.1.2**

The TSF shall store and protect certificate(s) from unauthorized deletion and modification.

**FIA_X509_EXT.1.3**

The TSF shall provide the capability for Authorized Administrators to load X.509v3 certificates into the TOE for use by the security functions specified in this PP.

**Component Assurance Activity:**

In order to show that the TSF supports the use of X.509v3 certificates according to the RFC 5280, the evaluator shall ensure that the TSS describes the following information:

For each section of RFC 5280, any statement that is not 'MUST' (for example, 'MAY', 'SHOULD', 'SHOULD NOT', etc.) shall be described so that the reader can determine whether the TOE implements that specific part of the standard;

For each section of RFC 5280, any non-conformance to 'MUST' or 'SHOULD' statements shall be described;

Any TOE-specific extensions or processing that is not included in the standard that may impact the security requirements the TOE is to enforce shall be described.

Additionally, the evaluator shall devise tests that show that the TOE processes certificates that conform to the implementation described in the TSS; are able to form a certification path as specified in the standard and in the TSS; and are able to validate certificates as specified in the standard (certification path validation including CRL processing). This testing shall be described in the team test plan.

It should be noted that future versions of this PP will have more explicit testing requirements for a TOE's certificate handling capability. Additionally, protocol-specific certificate handling testing will need to be performed and can be combined with the testing required by this assurance activity.

The TSS shall describe all certificate stores implemented that contain certificates used to meet the requirements of this PP. This description shall contain information pertaining to how certificates are loaded into the store, and how the store is protected from unauthorized access.

The evaluator shall perform the following tests for each function in the system that requires the use of certificates:

Test 1: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.

## 5.2.5 Security management (FMT)

### 5.2.5.1 Management of Security Functions Behavior (FMT_MOF.1)

**FMT_MOF.1.1**

Refinement: The TSF shall restrict the ability to enable, disable, determine and modify the behavior of all of the security functions of the TOE identified in this PP to the Authorized Administrator.

**Component Assurance Activity:**

The evaluator shall review the operational guidance to determine that each of the functions implemented in response to the requirements of this PP is identified, and that configuration information is provided to ensure that only administrators have access to the functions. The evaluator shall include in this list of functions to be examined those mechanisms dealing with adding additional instances of a TOE to a configuration, and configuration of the multiple TOE instances into a management hierarchy and/or redundant architecture. The evaluator shall examine the TSS to determine that, for each administrative function identified in the operational guidance, those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the configuration of the system through this interface is disallowed for non-administrative users.

### 5.2.5.2 Management of TSF Data (General TSF Data) (FMT_MTD.1(1))

**FMT_MTD.1.1(1)**

The TSF shall restrict the ability to manage the TSF data to the Authorized Administrators.

**Component Assurance Activity:**

Since administrative functions manipulate the TSF data, the analysis performed by the evaluators in the Assurance Activity for FMT_MOF.1 will demonstrate that this requirement is met.

#### 5.2.5.3 Management of TSF Data (Reading of Authentication Data) (FMT_MTD.1(2))

**FMT_MTD.1.1(2)**

Refinement: The TSF shall prevent reading of the password-based authentication data.

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and how they are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If passwords or other authentication data are not stored in plaintext, the TSS shall describe how the passwords are protected and how they are able to be used (e.g., administrator-entered passphrase).

#### 5.2.5.4 Management of TSF Data (for reading of all symmetric keys) (FMT_MTD.1(3))

**FMT_MTD.1.1(3)**

Refinement: The TSF shall prevent reading of all pre-shared keys, symmetric key, and private keys.

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that it details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

#### 5.2.5.5 Specification of management functions (FMT_SMF.1)

**FMT_SMF.1.1**

The TSF shall be capable of performing the following security management functions:

- Ability to configure the list of TOE services available before an entity is identified and authenticated, as specified in FIA_UIA.1, respectively.
- Ability to configure the cryptographic functionality.
- Ability to update the TOE, and to verify the updates using the digital signature capability (FCS_COP.1(2)) and
- No other functions.
- Ability to configure the TOE advisory notice and consent warning message regarding unauthorized use of the TOE.
- Ability to configure all security management functions identified in other sections of this PP.

**Component Assurance Activity:**

This requirement merely ensures that the mechanisms called for in other requirements are actually instantiated in the TOE; therefore, verification that these mechanisms exist and work in a manner consistent with the other requirements is provided through the Assurance Activities associated with those other requirements.

#### 5.2.5.6 Security Management Roles (FMT_SMR.1)

**FMT_SMR.1.1**

The TSF shall maintain the roles: Authorized Administrator; [No other roles].

**FMT_SMR.1.2**

The TSF shall be able to associate users with roles.

**FMT_SMR.1.3**

The TSF shall ensure that the conditions Authorized Administrator role shall be able to administer the TOE locally; Authorized Administrator role shall be able to administer the TOE remotely; he

ability to remotely administer the TOE remotely from a wireless client shall be disabled by default; are satisfied.

**Component Assurance Activity:**

The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

The evaluator shall also perform the following test:

Test 1: The evaluator shall demonstrate that after configuring the TOE for first use from the operational guidance, it is possible to establish an administrative session with the TOE on the 'wired' portion of the device. They shall then demonstrate that an identically configured wireless client that can successfully connect to the TOE cannot be used to perform administration.

## 5.2.6 Protection of the TSF (FPT)

### 5.2.6.1 Fail Secure (FPT_FLS.1)

**FPT_FLS.1.1**

The TSF shall preserve a secure state when the following types of failures occur: failure of the power-on self-tests.

**Component Assurance Activity:**

The evaluator shall review the TSS section to determine that the TOE's implementation of the fail secure functionality is documented. The evaluator shall first examine the TSS section to ensure that all failure modes specified in the ST are described. The evaluator shall then ensure that the TOE will attain a secure state after inserting each specified failure mode type. The evaluator shall review the TSS to determine that the definition of secure state is defined and is suitable to ensure protection of key material and user data.

### 5.2.6.2 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

**FPT_ITT.1.1**

Refinement: The TSF shall protect TSF data from disclosure and protect it from modification when it is transmitted between separate parts of the TOE through the use [***IPsec***].

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the communication paths for each supported method. The evaluator shall also perform the following tests:

Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) communications method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.

Test 2: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext.

Test 3: The evaluator shall ensure, for each method of communication, modification of the channel data is detected by the TOE.\par\line Further assurance activities are associated with the specific protocols.

#### 5.2.6.3 Replay Detection (FPT_RPL.1)

**FPT_RPL.1.1**

The TSF shall detect replay for the following entities: [network packets terminated at the TOE].

**FPT_RPL.1.2**

The TSF shall perform: [reject the data] when replay is detected.

#### 5.2.6.4 Reliable Time Stamp (FPT_STM.1)

**FPT_STM.1.1**

The TSF shall be able to provide reliable time stamps for its own use.

#### 5.2.6.5 Extended: TSF Testing (FPT_TST_EXT.1)

**FPT_TST_EXT.1.1**

The TSF shall run a suite of self-tests during the initial start-up (on power on) to demonstrate the correct operation of the TSF.

**FPT_TST_EXT.1.2**

The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic service specified in FCS_COP.1(2).

**Component Assurance Activity:**

The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying

#### 5.2.6.6 Extended: Trusted Update (FPT_TUD_EXT.1)

**FPT_TUD_EXT.1.1**

The TSF shall provide authorized administrators the ability to query the current version of the TOE firmware/software.

**FPT_TUD_EXT.1.2**

The TSF shall provide authorized administrators the ability to initiate updates to TOE firmware/software.

**FPT_TUD_EXT.1.3**

The TSF shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [***published hash***] prior to installing those updates.

**Component Assurance Activity:**

Updates to the TOE are signed by an authorized source and may have a hash associated. For the digital signature mechanism, the definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature, and if implemented, calculating the hash of the updates; and the actions that take place for successful (signature, and hash if included, verifications) and unsuccessful (signature, and hash if included. could not be verified) cases. The evaluator shall perform the following tests:

Test 1: The evaluator performs the version verification activity to determine the current version of

the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator performs a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update.

Test 2: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TOE rejects the update.

## 5.2.7 Resource utilisation (FRU)

### 5.2.7.1 Maximum Quotas TOE Access (FTA) (FRU_RSA.1)

**FRU_RSA.1.1**

The TSF shall enforce maximum quotas of the following resources: [**control-plane bandwidth**], [***no other resources***] that [***individual users***] can use [***simultaneously***].

**Component Assurance Activity:**

The evaluator shall examine the TSS to ensure that it identifies all resources controlled through the quota mechanism, and that this list contains those resources used to support the administrative interface. The evaluator shall ensure that the TSS describes how each resource is counted as 'used' and how a maximum quota or use is determined, as well as the action taken when the quota is reached. The TSS shall also describe whether the quota is imposed on users or subjects (in this case TOE processes) and whether the quota imposed is for simultaneous use or cumulative use over a period of time. The evaluator shall examine the operational guidance to determine that it contains instructions for establishing quotas (if they are configurable), and describes any actions administrators can or should take in response to a quota being reached.

The evaluator shall also perform the following tests for each controlled resource:

Test 1: The evaluator follows the operational guidance to configure quotas for the resource (if such a capability is provided). The evaluator then causes the resource quota to be reached, and observes that the action specified in the TSS occurs.

## 5.2.8 TOE access (FTA)

### 5.2.8.1 TSF-initiated termination (FTA_SSL.3)

**FTA_SSL.3.1**

The TSF shall terminate a remote interactive session after an Authorized Administrator-configurable time interval of session inactivity.

**Component Assurance Activity:**

The evaluator shall perform the following test:

Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component; these shall consist at least of the minimum and maximum allowed values as specified in the operational guidance, as well as one other value. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.

### 5.2.8.2 User-initiated termination (FTA_SSL.4)

**FTA_SSL.4.1**

The TSF shall allow Administrator-initiated termination of the Administrator’s own interactive session.

**Component Assurance Activity:**

The evaluator shall perform the following test:

Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.

Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.

#### 5.2.8.3 TSF-initiated session locking (FTA_SSL_EXT.1)

**FTA_SSL_EXT.1.1**

Refinement: The TSF shall, for local interactive sessions, [***terminate the session***] after an Authorized Administrator specified time period of inactivity.

**Component Assurance Activity:**

The evaluator shall perform the following test:

Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

#### 5.2.8.4 Default TOE Access Banners (FTA_TAB.1)

**FTA_TAB.1.1**

Refinement: Before establishing an administrative user session the TSF shall be capable of displaying an Authorized Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE.

**Component Assurance Activity:**

The evaluator shall check the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS). The evaluator shall also perform the following test:

Test 1: The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

#### 5.2.8.5 TOE Session Establishment Trusted Path/Channels (FTP) (FTA_TSE.1)

**FTA_TSE.1.1**

Refinement: The TSF shall be able to deny establishment of a wireless client session based on location, time, day, [**blacklist state**].

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that all of the attributes on which a client session can be denied are specifically defined. The evaluator shall examine the operational guidance to determine that it contains guidance for configuring each of the attributes identified in the TSS. The evaluator shall also perform the following test for each attribute:

Test 1: The evaluator successfully establishes a client session with a wireless client. The evaluator then follows the operational guidance to configure the system so that that client's access is denied

based on a specific value of the attribute. The evaluator shall then attempt to establish a session in contravention to the attribute setting (for instance, the location is denied based upon the client's IP address). The evaluator shall observe that the access attempt fails.

## 5.2.9 Trusted path/channels (FTP)

### 5.2.9.1 Inter-TSF trusted channel (FTP_ITC.1)

**FTP_ITC.1.1**

Refinement: The TSF shall use 802.11-2007, IPsec, and [***no other protocols***] to provide a trusted communication channel between itself and all authorized IT entities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

**FTP_ITC.1.2**

The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

**FTP_ITC.1.3**

The TSF shall initiate communication via the trusted channel for [**remote logging, NTP, and authentication functions**].

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken. The evaluator shall also perform the following tests:

Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.

Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.

Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

Test 4: The evaluator shall ensure, for each communication channel with an authorized IT entity, modification of the channel data is detected by the TOE.

Test 5: The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

Further assurance activities are associated with the specific protocols.

### 5.2.9.2 Trusted Path (FTP_TRP.1)

**FTP_TRP.1.1**

Refinement: The TSF shall use [***SSH, TLS/HTTPS***] provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.

**FTP_TRP.1.2**

Refinement: The TSF shall permit remote administrators to initiate communication via the trusted path.

**FTP_TRP.1.3**

The TSF shall require the use of the trusted path for initial administrator authentication and all remote administration actions.

**Component Assurance Activity:**

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator shall also perform the following tests:

Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.

Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative session without invoking the trusted path.

Test 3: The evaluator shall ensure, for each method of remote administration, the channel data is not sent in plaintext.

Test 4: The evaluator shall ensure, for each method of remote administration, modification of the channel data is detected by the TOE.

Further assurance activities are associated with the specific protocols.

## 5.3 TOE Security Assurance Requirements

The SARs for the TOE are the EAL 1 components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

| Requirement Class | Requirement Component |
|---|---|
| **ADV: Development** | ADV_FSP.1: Basic functional specification |
| **AGD: Guidance documents** | AGD_OPE.1: Operational user guidance |
| | AGD_PRE.1: Preparative procedures |
| **ALC: Life-cycle support** | ALC_CMC.1: Labelling of the TOE |
| | ALC_CMS.1: TOE CM coverage |
| **ATE: Tests** | ATE_IND.1: Independent testing - conformance |
| **AVA: Vulnerability assessment** | AVA_VAN.1: Vulnerability survey |

**Table 3 EAL 1 Assurance Components**

## 5.3.1 Development (ADV)

### 5.3.1.1 Basic functional specification (ADV_FSP.1)

**ADV_FSP.1.1d**

The developer shall provide a functional specification.

**ADV_FSP.1.2d**

The developer shall provide a tracing from the functional specification to the SFRs.

**ADV_FSP.1.1c**

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

**ADV_FSP.1.2c**

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

**ADV_FSP.1.3c**

The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.

**ADV_FSP.1.4c**

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

**ADV_FSP.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**ADV_FSP.1.2e**

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

**Component Assurance Activity:**

There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 4.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.

## 5.3.2 Guidance documents (AGD)

### 5.3.2.1 Operational user guidance (AGD_OPE.1)

**AGD_OPE.1.1d**

The developer shall provide operational user guidance.

**AGD_OPE.1.1c**

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

**AGD_OPE.1.2c**

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

**AGD_OPE.1.3c**

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

**AGD_OPE.1.4c**

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

**AGD_OPE.1.5c**

The operational user guidance shall identify all possible modes of operation of the TOE (including

operation following failure or operational error), their consequences and implications for maintaining secure operation.

**AGD_OPE.1.6c**

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

**AGD_OPE.1.7c**

The operational user guidance shall be clear and reasonable.

**AGD_OPE.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**Component Assurance Activity:**

During operation, the activities to be described in the guidance fall into two broad categories; those that are performed by a (non-administrative) user, and those that are performed by an administrator. It should be noted that most procedures needed for non-administrative users are referenced in the assurance activities in Section 4.1.

With respect to the administrative functions, while several have also been described in Section 4.1, additional information is required as follows.

The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely more than one of these, and this is not limited to the process that.

#### 5.3.2.2 Preparative procedures (AGD_PRE.1)

**AGD_PRE.1.1d**

The developer shall provide the TOE including its preparative procedures.

**AGD_PRE.1.1c**

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

**AGD_PRE.1.2c**

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

**AGD_PRE.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**AGD_PRE.1.2e**

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

**Component Assurance Activity:**

As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms and components (that is, combination of hardware and operating system) claimed for the TOE in the ST.

The evaluator shall check to ensure that the following guidance is provided:

Instructions and information is provided to the administrator detailing how to configure the virtual management network so that control/configuration network traffic between TOE components is encrypted and that this is the only allowed configuration for conformant TOEs. If the TOE is a multiple component TOE, then the appropriate requirements are included in the ST from

Appendix C and the assurance activities associated with those requirements provide details on the guidance necessary for both the TOE and operational environment.

As indicated in the introductory material, administration of the TOE is performed by administrator role. At a high level, the guidance must contain the appropriate instructions to allow local and remote authenticated administrator access.

## 5.3.3 Life-cycle support (ALC)

### 5.3.3.1 Labelling of the TOE (ALC_CMC.1)

**ALC_CMC.1.1d**

The developer shall provide the TOE and a reference for the TOE.

**ALC_CMC.1.1c**

The TOE shall be labelled with its unique reference.

**ALC_CMC.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**Component Assurance Activity:**

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

### 5.3.3.2 TOE CM coverage (ALC_CMS.1)

**ALC_CMS.1.1d**

The developer shall provide a configuration list for the TOE.

**ALC_CMS.1.1c**

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

**ALC_CMS.1.2c**

The configuration list shall uniquely identify the configuration items.

**ALC_CMS.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**Component Assurance Activity:**

The 'evaluation evidence required by the SARs' in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.

## 5.3.4 Tests (ATE)

### 5.3.4.1 Independent testing - conformance (ATE_IND.1)

**ATE_IND.1.1d**

The developer shall provide the TOE for testing.

**ATE_IND.1.1c**

The TOE shall be suitable for testing.

**ATE_IND.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**ATE_IND.1.2e**

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

**Component Assurance Activity:**

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the body of this PP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluators must document in the test plan that each applicable testing requirement in the ST is covered.

The Test Plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platform and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluators are expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) is provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform.

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a 'fail' and 'pass' result (and the supporting details), and not just the 'pass' result.

## 5.3.5 Vulnerability assessment (AVA)

### 5.3.5.1 Vulnerability survey (AVA_VAN.1)

**AVA_VAN.1.1d**

The developer shall provide the TOE for testing.

**AVA_VAN.1.1c**

The TOE shall be suitable for testing.

**AVA_VAN.1.1e**

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

**AVA_VAN.1.2e**

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

**AVA_VAN.1.3e**

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

**Component Assurance Activity:**

As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to

determine the vulnerabilities that have been found in WLAN Access System products in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, for example, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires an electron microscope and a tank of liquid nitrogen, for instance, then a test would not be suitable and an appropriate justification would be formulated.

# 6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Resource utilisation
- TOE access
- Trusted path/channels

Note that some Assurance Activities require information about design and implementation choices made in regard to RFCs. The applicable information is included in tabular form at the end of this document in section 8.6.

## 6.1 Security audit

The TOE has an audit generation mechanism to record security and non-security relevant events. There are several types of category for audit logs including Network, System, Security, Wireless, and User. The Network log category can include all network packets, protocol packet dump, mobility, and DHCP events. The System log category can include all system, configuration, SNMP, and web server events. The Security log category can include all security, AAA, firewall, packet trace, VPN, 802.1x, and IKE events. The Wireless log category can include all wireless events. The User log category can include all user, VPN, 802.1X, and RADIUS user events.

The events that can cause an audit record to be logged include starting and stopping the audit function, any use of an administrator command via the CLI and Web interfaces, as well as all of the events identified in **Table 2 Audit Events**.

The administrator can turn on or off (include or exclude) auditable events based on specific criteria. The administrator can configure the logging level (event type) for each of the modules (AP, network, security, system, user and wireless; no related to software modules) of the ArubaOS. Please note that only the MC generates audit events. The inclusion and exclusion of audited events for the event type is performed by using the "logging" command at the CLI. The Web GUI provides similar functionality through the *Monitoring->Management->Logging* panel. There are a total of eight syslog logging levels and the default logging level for all categories is Warning. Set the logging level to "Warning" for all Categories and Subcategories to generate all of the security event logs as defined in FAU_GEN.1 Table 2. The logging levels are defined as followed:

| | |
|---|---|
| Emergency | Panic conditions that occur when the system becomes unusable. |
| Alert | Any condition requiring immediate attention and correction. |
| Critical | Any critical conditions such as a hard drive error. |
| Errors | Error conditions |
| Warning | Warning messages |
| Notice | Significant events of a non-critical and normal nature. |
| Informational | Messages of general interest to system users. |
| Debug | Messages containing information useful for debugging. |

The TOE generates audit records for auditable events. The audit function is integrated into each module of ArubaOS. In particular, when an auditable event occurs, the module executes a logging API call that records event information to the external audit (syslog) server. The audit records are transmitted to the audit server over a trusted channel and are stored and protected by the audit server. At the audit server, the administrator is provided with an

interface (part of operating environment) to read audit logs. Though not required by PP, the TOE also stores audit records locally and provides CLI and WebUI capabilities to view the contents of the audit trail. The local cache for audit data is 3*(3*31768 bytes). Since this local protected log storage is FIFO (First in, First out), audit logs are overwritten when the storage is exhausted. For each audit event, the following minimum information is recorded:

| | |
|---|---|
| Event Type | The logging level |
| Subject Identity | The identity of the subject involved in the event. For identified users, the subject identity is represented by the username. For other subjects, the subject identity is represented by the IP address for wired network subjects and by the MAC address for wireless network subjects. |
| Date and Time | The date and time when each event occurred. The time can be obtained internally or from a trusted NTP server in the operating environment. |
| Outcome | Success or failure of the event |

The logged audit records also include event-specific content that includes at least all of the content required in **Table 2 Audit Events**.

The TOE provides functions allowing administrators to review all audit information stored on the TOE.

The TOE (MC and AP) can generate and send SNMP traps to the operating environment (SNMP server) to alert the administrators of potential problems, misconfiguration, or attacks.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: The TOE generates audit events for various purposes such as security and trouble shooting. The events include startup and shutdown of audit function, all authentication attempts, all administrative actions, and all required auditable events as specified in Table 2 Audit Events. At a minimum, each event includes date and time, logging level (event type), subject identity, and outcome of event.
- FAU_GEN.2: The TOE associates user id to the appropriate audit event. In other words, the user is identified by the username in the audit record.
- FAU_SAR.1: The TOE provides audit review functions allowing administrators to review all audit data stored by the TOE.
- FAU_SAR.2: The TOE provides access to stored audit records only to TOE administrators.
- FAU_SEL.1: The TOE provides administrators the capability to include or exclude audit events based on event type (implying outcome where applicable). The requirement can additionally be met with an external logging server configured to only capture information by user id (username, device interface (e.g., VLAN0, ETH1), and wireless client identity (MAC Address). The TOE takes a secure approach of auditing *all* information to ensure thorough analysis can be performed, if desired. Most logs generated on the TOE are not administrator-initiated and those that are deal directly with authentication to the TOE and configuration actions. An administrator can choose to selectively audit based upon event type, capturing only failing events, passing events, or all events. This is done by specifying the logging level for the specific event through the GUI or by command line via SSH.
- FAU_STG.1: While it is recommended to use an external server for audit data in the evaluated configuration, the local cache is 3*(3*31768 bytes).The local logs can only be viewed – they cannot be deleted or modified. There are no CLI commands for such actions. The only way to delete local audit records is to go outside the evaluated configuration and reset the controller to factory defaults, or to wipe the flash storage. If an external syslog server has been enabled, all audit logs are simultaneously written to both the local audit log and the syslog server. Local audit logs and logs sent to a remote server are identical.
- FAU_STG_EXT.1: The TOE will be configured to use IPsec when exporting audit records to an external SYSLOG server.
- FAU_STG_EXT.3:The TOE uses the BSD Syslog Protocol (RFC 3164) which operates over UDP and is purely connectionless. In order to detect a failure of the connection to the syslog server, the path must operate over an IPsec tunnel between the TOE and the syslog server. Failure of the IPsec tunnel will indirectly indicate failure of the audit server or the path to the audit server. A local log message will

indicate such a failure. An administrator must take action to manually re-synchronize the remote audit log after the path is restored.

## 6.2 Cryptographic support

The TOE meets FIPS 140-2 requirements by allowing the administrator to enable a FIPS operating mode. The CC evaluated configuration of the TOE requires the use of this FIPS operating mode. In this mode, only FIPS-approved algorithms are allowed for cryptographic services (e.g., encryption, hashing, digital signature, etc.). All use of cryptographic services (e.g., TLSv1, IPsec/IKE, SSHv2, etc.) can only utilize FIPS-approved algorithms for the underlying algorithms. All models are FIPS-certified at overall Level 2. This ensures that tamper-evident seals are placed around the enclosure (as specified by FIPS 140-2 requirements) to detect any tampering. In addition, at Level 2, any ventilation holes or slots must be small or obstructed to prevent probing of the inside.

The following functions have been FIPS certified in accordance with the identified standards.

| Functions | Standards | Certificates |
|---|---|---|
| Symmetric key derivation | | |
| • WPA2 128-bit cryptographic key derivation | PRF-384<br>802.11-2007 | Cert #426, 1512, 1663, 1666 |
| • Random number generation | ***NIST Special Publication 800-90 using [CTR_DRBG(any)*** | Cert #443 |
| Asymmetric key generation | | |
| • Domain parameter generation | NIST Special Publication 800-56A<br>NIST Special Publication 800-56B | Cert #466, 1376, 1380 |
| Key Distribution | | |
| • Pairwise Master Key reception from an 802.1X Authorization Server | 802.11-2007<br>IEEE 802.11i<br>Note: If RADIUS protocol is used between the Authenticator and AS is RADIUS. The MS-MPPE-Recv-Key attribute (vendor-id = 17; see Section 2.4.3 in IETF RFC 2548-1999 [B30]) is used to transport the PMK to the Authenticator. | N/A |
| • AES Key Wrap in an EAPOL-Key frame for GTK | RFC 3394 for AES Key Wrap, 802.11-2007 for the packet format and timing considerations | Cert #762, 779, 2479, 2680, 2677 |
| • Group Key Handshake | IEEE 802.11i | |
| Encryption/Decryption | | |
| • AES CCM and GCM (128-256 bits) | FIPS PUB 197<br>NIST SP 800-38C<br>NIST SP 800-38D | Cert #779, 1648, 2680, 2677 |
| • AES CCMP (128 bits) | FIPS PUB 197<br>NIST SP 800-38C<br>IEEE 802.11-2007 | Cert #779, 1648, 2680, 2677 |
| Cryptographic signature services | | |
| • RSA Digital Signature Algorithm (rDSA) (modulus 2048) | FIPS PUB 186-3 | Cert #1376, 1379, 1380 |
| • Elliptic Curve Digital Signature Algorithm (ECDSA) (P-256 and P-384) | FIPS PUB 186-3 | Cert #466, 469 |
| Cryptographic hashing | | |
| • SHA-1, SHA-256, and SHA-384 (digest sizes 160, 256, and 384 bits) | FIPS Pub 180-3 | Cert #762, 781, 934, 2246, 2249, 2250 |
| Keyed-hash message authentication | | |
| • HMAC-SHA-1, HMAC-SHA-256, | FIPS Pub 198-1 | Cert #417, 416, 538, |

| | | |
|---|---|---|
| HMAC-SHA-384, and SHA-1-96 (digest sizes 160, 256, and 384 bits) | FIPS Pub 180-3 | 1663, 1666 |
| Random bit generation | | |
| • RGB with one independent hardware based noise source of 256 bits of non-determinism | ***NIST Special Publication 800-90 using [CTR_DRBG(any)*** | Cert #433, 1250 |

**Table 4 Cryptographic Functions**

While the TOE generally fulfills all of the NIST SP 800-56A and 800-56B requirements without extensions, the following tables specifically identify the "should", "should not", and "shall not" conditions from those publications along with an indication of how the TOE conforms to those conditions.

| NIST SP800-56A Section Reference | Requirement/ Recommendation Qualifier | Implemented? | Rationale for deviation |
|---|---|---|---|
| 5.4 | should | yes | Not applicable |
| 5.5.1.1 | should | yes | Not applicable |
| 5.5.2 | should | yes | Not applicable |
| 5.6.2 | should | yes | Not applicable |
| 5.6.2.1 | should | yes | Not applicable |
| 5.6.2.2 | should | yes | Not applicable |
| 5.6.2.3 | should | yes | Not applicable |
| 5.6.3.1 | should | yes | Not applicable |
| 5.6.3.2.1 | should | yes | Not applicable |
| 5.6.4.1 | shall not | no | Not applicable |
| 5.6.4.2 | should | yes | Not applicable |
| 5.6.4.2 | shall not | no | Not applicable |
| 5.6.4.3 | should (first occurrence) | yes | Not applicable |
| 5.6.4.3 | should (second occurrence) | yes | Not applicable |
| 5.8 | shall not (first occurrence) | no | Not applicable |
| 5.8 | shall not (second occurrence) | no | Not applicable |
| 6 | should | yes | Not applicable |
| 7 | shall not (first occurrence) | no | Not applicable |
| 7 | shall not (second occurrence) | no | Not applicable |
| 9 | shall not | no | Not applicable |

**Table 5 NIST SP800-56A Conformance**

| NIST SP800-56B Section Reference | "should", "should not", or "shall not" | Implemented? | Rationale for deviation |
|---|---|---|---|
| 5.6 | should | yes | Not applicable |
| 5.8 | shall not | no | Not applicable |
| 5.9 | shall not (first occurrence) | no | Not applicable |
| 5.9 | shall not (second occurrence) | no | Not applicable |
| 6.1 | should not | no | Not applicable |
| 6.1 | should (first occurrence) | yes | Not applicable |
| 6.1 | should (second occurrence) | yes | Not applicable |
| 6.1 | should (third occurrence) | yes | Not applicable |
| 6.1 | should (fourth occurrence) | yes | Not applicable |
| 6.1 | shall not (first occurrence) | no | Not applicable |
| 6.1 | shall not (second occurrence) | no | Not applicable |
| 6.2.3 | should | yes | Not applicable |

| NIST SP800-56B Section Reference | "should", "should not", or "shall not" | Implemented? | Rationale for deviation |
|---|---|---|---|
| 6.5.1 | should | yes | Not applicable |
| 6.5.2 | should | yes | Not applicable |
| 6.5.2.1 | should | yes | Not applicable |
| 6.6 | shall not | no | Not applicable |
| 7.1.2 | should | yes | Not applicable |
| 7.2.1.3 | should | yes | Not applicable |
| 7.2.1.3 | should not | no | Not applicable |
| 7.2.2.3 | should (first occurrence) | yes | Not applicable |
| 7.2.2.3 | should (second occurrence) | yes | Not applicable |
| 7.2.2.3 | should (third occurrence) | yes | Not applicable |
| 7.2.2.3 | should (fourth occurrence) | yes | Not applicable |
| 7.2.2.3 | should not | no | Not applicable |
| 7.2.2.3 | shall not | no | Not applicable |
| 7.2.3.3 | should (first occurrence) | yes | Not applicable |
| 7.2.3.3 | should (second occurrence) | yes | Not applicable |
| 7.2.3.3 | should (third occurrence) | yes | Not applicable |
| 7.2.3.3 | should (fourth occurrence) | yes | Not applicable |
| 7.2.3.3 | should (fifth occurrence) | yes | Not applicable |
| 7.2.3.3 | should not | no | Not applicable |
| 8 | should | yes | Not applicable |
| 8.3.2 | should not | no | Not applicable |

**Table 6 NIST SP800-56B Conformance**

| Name | CSPs type | Generation | Storage and Zeroization | Use |
|---|---|---|---|---|
| Key Encryption Key (KEK) | Triple-DES 168-bit key | Hardcoded during manufacturing | Stored in Flash. Zeroized by using command 'wipe out flash' | Encrypts IKEv1/IKEv2 Pre-shared key, RADIUS server shared secret, RSA private key, ECDSA private key, 802.11i pre-shared key and Passwords. |
| DRBG entropy input | SP800-90a DRBG (512 bits) | Derived using NON-FIPS approved HW RNG | Stored in plaintext in volatile memory. Zeroized on reboot. | DRBG initialization |
| DRBG seed | SP800-90a DRBG (384 bits) | Generated per SP800-90A using a derivation function | Stored in plaintext in volatile memory. Zeroized on reboot. | DRBG initialization |

| | | | | |
|---|---|---|---|---|
| DRBG Key | SP800-90a (256 bits) | Generated per SP800-90A | Stored in plaintext in volatile memory. Zeroized on reboot. | DRBG |
| DRBG V | SP800-90a (128 bits) | Generated per SP800-90A | Stored in plaintext in volatile memory. Zeroized on reboot. | DRBG |
| RNG seed | FIPS 186-2 RNG Seed (512 bits) | Derived using NON-FIPS approved HW RNG | Stored in plaintext in volatile memory. Zeroized on reboot. | Seed 186-2 General purpose (x-change Notice); SHA-1 RNG |
| RNG seed key | FIPS 186-2 RNG Seed key (512 bits) | Derived using NON-FIPS approved HW RNG | Stored in plaintext in volatile memory. Zeroized on reboot. | Seed 186-2 General purpose (x-change Notice); SHA-1 RNG |
| Diffie-Hellman private key | Diffie-Hellman private key (160/224 bits) | Generated internally during Diffie-Hellman Exchange | Stored in the volatile memory. Zeroized after the session is closed. | Used in establishing the session key for an IPSec session |
| Diffie-Hellman public key | Diffie-Hellman public key (1024/2048 bits) Note: Key size of DH Group 1 (768 bits) is not allowed in FIPS mode. | Generated internally during Diffie-Hellman Exchange | Stored in the volatile memory. Zeroized after the session is closed. | Used in establishing the session key for an IPSec session |
| Diffie-Hellman shared secret | Diffie-Hellman shared secret (1024/2048 bits) | Established during Diffie-Hellman Exchange | Stored in plain text in volatile memory, Zeroized when session is closed. | Key agreement in SSHv2 |
| EC Diffie-Hellman private key | Elliptic Curve Diffie-Hellman (P-256 and P-384). | Generated internally during EC Diffie-Hellman Exchange | Stored in the volatile memory. Zeroized after the session is closed. | Used in establishing the session key for an IPSec session |
| EC Diffie-Hellman public key | Elliptic Curve Diffie-Hellman (P-256 and P-384). | Generated internally during EC Diffie-Hellman Exchange | Stored in the volatile memory. Zeroized after the session is closed. | Used in establishing the session key for an IPSec session |

| | | | | |
|---|---|---|---|---|
| EC Diffie-Hellman shared secret | Elliptic Curve Diffie-Hellman ( P-256 and P-384) | Established during EC Diffie-Hellman Exchange | Stored in plaintext in volatile memory. Zeroized when session is closed. | Key agreement in IKEv1/IKEv2 |
| RADIUS server shared secret | 8-128 character shared secret | CO configured | Stored encrypted in Flash with the KEK. Zeroized by changing (updating) the pre-shared key through the User interface. | Module and RADIUS server authentication |
| Enable secret | 8-64 character password | CO configured | Store in ciphertext in flash. Zeroized by changing (updating) through the user interface. | Administrator authentication |
| User Passwords | 8-64 character password | CO configured | Stored encrypted in Flash with KEK. Zeroized by either deleting the password configuration file or by overwriting the password with a new one. | Authentication for accessing the management interfaces, RADIUS authentication |
| IKEv1/IKEv2 Pre-shared key | 64 character pre-shared key | CO configured | Stored encrypted in Flash with the KEK. Zeroized by changing (updating) the pre-shared key through the User interface. | User and module authentication during IKEv1, IKEv2 |
| skeyid | HMAC-SHA-1/256/384 (160/256/384 bits) | Established during IKEv1 negotiation | Stored in plaintext in volatile memory. Zeroized when session is closed. | Key agreement in IKEv1 |
| skeyid_d | HMAC-SHA-1/256/384 (160/256/384 bits) | Established during IKEv1 negotiation | Stored in plaintext in volatile memory. Zeroized when session is closed. | Key agreement in IKEv1 |
| IKEv1/IKEv2 session authentication key | HMAC-SHA-1/256/384 (160 / 256 / 384 bits) | Established as a result of IKEv1/IKEv2 service implementation. | Stored in plaintext in volatile memory. Zeroized when session is closed. | IKEv1/IKEv2 payload integrity verification |
| IKEv1/IKEv2 session encryption key | Triple-DES (168 bits/AES (128/196/256 bits) | Established as a result of IKEv1/IKEv2 service implementation. | Stored in plaintext in volatile memory. Zeroized when session is closed. | IKEv1/IKEv2 payload encryption |

| | | | | |
|---|---|---|---|---|
| IPSec session encryption keys | Triple-DES (168 bits / AES (128/196/256 bits) | Established during the IPSec service implementation | Stored in plaintext in volatile memory. Zeroized when the session is closed. | Secure IPSec traffic |
| IPSec session authentication keys | HMAC-SHA-1 (160 bits) | Established during the IPSec service implementation | Stored in plaintext in volatile memory. Zeroized when the session is closed. | User authentication |
| SSHv2 session keys | AES (128/196/256 bits) | Established during the SSHv2 key exchange | Stored in plaintext in volatile memory. Zeroized when the session is closed. | Secure SSHv2 traffic |
| SSHv2 session authentication key | HMAC-SHA-1 (160-bit) | Established during the SSHv2 key exchange | Stored in plaintext in volatile memory. Zeroized when the session is closed. | Secure SSHv2 traffic |
| TLS pre-master secret | 48 byte secret | Externally generated | Stored in plaintext in volatile memory. Zeroized when the session is closed. | TLS key agreement |
| TLS session encryption key | AES 128/192/256 bits | Generated in the module during the TLS service implementation | Stored in plaintext in volatile memory. Zeroized when the session is closed. | TLS session encryption |
| TLS session authentication key | HMAC-SHA-1/256/384 (160/256/384 bits) | Generated in the module during the TLS service implementation | Stored in plaintext in volatile memory. Zeroized when the session is closed. | TLS session authentication |
| RSA Private Key | RSA 1024/2048 bit private key | Generated in the module | Stored in flash memory encrypted with KEK. Zeroized by the CO command **write erase all**. | Used by TLS and EAP-TLS/PEAP protocols during the handshake, used for signing OCSP responses, and used by IKEv1/IKEv2 for device authentication and for signing certificates |
| RSA public key | RSA 1024/2048 bit public key | Generated in the module | Stored in flash memory encrypted with KEK. Zeroized by the CO command **write erase all**. | Used by TLS and EAP-TLS/PEAP protocols during the handshake, used for signing OCSP responses, and used by IKEv1/IKEv2 for device authentication and for signing certificates |

| | | | | |
|---|---|---|---|---|
| ECDSA Private Key | ECDSA suite B P-256 and P-384 curves | Generated in the module | Stored in flash memory encrypted with KEK. Zeroized by the CO command **write erase all**. | Used by TLS and EAP-TLS/PEAP protocols during the handshake. |
| ECDSA Public Key | ECDSA suite B P-256 and P-384 curves | Generated in the module | Stored in flash memory encrypted with KEK. Zeroized by the CO command **write erase all**. | Used by TLS and EAP-TLS/PEAP protocols during the handshake. |
| 802.11i Pre-Shared Key (PSK) | 8-63 character 802.11i pre-shared secret for use in 802.11i (SP 800-108) key derivation | CO configured | Stored in flash memory encrypted with KEK. Zeroized by the CO command **write erase all**. | Used by the 802.11i protocol |
| 802.11i Pair-Wise Master key (PMK) | 802.11i secret key (256-bit) | Derived during the EAP-TLS/PEAP handshake | Stored in the volatile memory. Zeroized on reboot. | Used by the 802.11i protocol |
| 802.11i session key | AES-CCM key (128 bits), AES-GCM key (128/256 bits) | Derived from 802.11 PMK | Stored in plaintext in volatile memory. Zeroized on reboot. | Used for 802.11i encryption |
| SNMPv3 authentication password | 8-64 character password | CO configured | Stored in flash memory encrypted with KEK. Zeroized by the CO command **write erase all.** | Used for SNMPv3 authentication |
| SNMPv3 privacy password | 8-64 character password | CO configured | Stored in flash memory encrypted with KEK. Zeroized by the CO command **write erase all.** | Used to derive SNMPv3 session key |
| SNMPv3 session key | AES-CFB key (128 bits) | Derived from SNMPv3 privacy password using an approved KDF | Stored in volatile memory. Zeroized on reboot. | Secure channel for SNMPv3 management |

**Table 7**

Entropy source (FCS_RBG_EXT.1.2) is described in a separate document, "Aruba Mobility Controller Entropy Documentation" prepared according to NDPP Annex D. To TOE uses its RBG capability to generate an "x" for each of its DH groups, For IKEv1 the TOE makes a call to OpenSSL's DRBG and for IKEv2 via a call to the ArubaOS Crypto Module (Mocana) FIPS186 RNG. Similarly, the TOE generates nonces such that the probability a specific nonce value will be repeated during the lifetime of a specific IPsec SA satisfies the restrictions specified in FCS_IPSEC_EXT.1.6 with IKEv1 calling DRBG and IKEv2 calling RNG.

Additionally, the TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. This function has also been subject to FIPS 140 certification. Zeroization is accomplished by overwriting the secret or private key with all zeroes.

The supporting cryptographic functions are included to support the HTTPS/TLS (RFCs 2818 TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246), SSHv2 (RFCs 4251, 4252, 4253, and 4254), and IPsec (RFC 4303) secure communication protocol.

The following cipher suites are implemented by the TOE:

- TLS_RSA_WITH_AES_128_CBC_SHA,
- TLS_RSA_WITH_AES_256_CBC_SHA,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA;
- TLS_RSA_WITH_AES_128_CBC_SHA256,
- TLS_RSA_WITH_AES_256_CBC_ SHA256,
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

The TOE supports SSHv2 with AES (CBC) 128 or 256 bit ciphers, in conjunction with HMAC-SHA-1 or HMAC-SHA-1-96, and RSA (with diffie-hellman-group14-sha1 for the key exchange method). In the FIPS mode of operation, the cipher parameters have been hardcoded to use only the Common Criteria evaluated configuration and are not configurable by the administrator.

SSHv2 connections are rekeyed prior to reaching $2^{28}$ packets; the authentication timeout period is 30 seconds allowing clients to retry only 3 times, password based authentication can be configured; and packets are limited to 32,768 bytes. Note that the TOE manages a packet counter for each SSH session so that it can initiate a new key exchange when the $2^{28}$ packet limit is reached. Whenever the timeout period or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (32,768 bytes) the packet will be dropped.

The TOE includes an implementation of IPsec in accordance with RFC 4303 for security. The primary cryptographic algorithms used by the TOE include AES-CBC-128 and AES-CBC-256 (as specified by RFC 3602) and AES-GCM-128 and AES-GCM-256 (as specified by RFC 4106) along with IKEv1 as defined in RFCs 2407, 2408, 2409, and RFC 4109 and IKE2 as defined in RFCs 5996 and 4307. Note that the TOE supports both main and aggressive modes, though aggressive mode is disabled in CC/FIPS mode as indicated above. Furthermore, "confidentiality only" ESP mode is disabled by default.

IKEv1 and IKEv2 SA lifetime and volume limits can be configured via a CLI function by an authorized administrator and, in the case of IKEv1, can be limited to 24 hours for phase 1 and 8 hours for phase 2 and also to as little as 2.5 MB of traffic for phase 2. The IKEv1 and IKEv2 protocols implemented by the TOE include DH Groups 2 (1024-bit MODP), 14 (2048-bit MODP), 19 (256-bit ECC), and 20 (384-bit ECC) and utilize RSA (aka rDSA) or ECDSA peer authentication. In the IKEv1 phase 1 and phase 2 and IKEv2 IKE_SA and IKE_CHILD exchanges, the TOE and peer will agree on the best DH group both can support. When the TOE initiates IKE negotiation, the DH group is sent in order according to the peer's configuration. When the TOE receives an IKE proposal, it will select the first match and the negotiation will fail if there is no match. The TOE does not execute any checks to ensure the strength of the negotiated symmetric algorithm in the IKEv1Phase 2SA or IKEv2 CHILD_SA is less than or equal of the strength of the IKEv1 Phase 1 SA or IKEv2 IKE_SA. As such, it is the responsibility of the Administrator to properly configure quick mode 1+2 for algorithms.

Additionally, pre-shared keys used for IPsec can be constructed of essentially any alphabetic character (upper and lower case), numerals, and special characters (e.g., "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")") and can be anywhere from 1 to 64 bytesin length (e.g., 22 characters). The TOE requires suitable keys to be entered by an authorized administrator using a Web GUI or CLI function.

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1(1): See table above.
- FCS_CKM.1(2): See table above.
- FCS_CKM.2(1): See table above.
- FCS_CKM.2(2): See table above.
- FCS_CKM_EXT.4: Keys are zeroized when they are no longer needed by the TOE
- FCS_COP.1(1): See table above.
- FCS_COP.1(2): See table above.
- FCS_COP.1(3): See table above.
- FCS_COP.1(4): See table above.
- FCS_COP.1(5): See table above.
- FCS_HTTPS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions.
- FCS_IPSEC_EXT.1: The TOE supports IPsec cryptographic network communication protection (RFC 4868, RFC 4945).
- FCS_RBG_EXT.1: See table above.
- FCS_SSH_EXT.1: The TOE supports SSHv2 interactive command-line secure administrator sessions as indicated above.
- FCS_TLS_EXT.1: The TOE supports TLS/HTTPS web-based secure administrator sessions.

## 6.3 User data protection

Network packets are received in memory buffers pre-allocated at boot time. The buffers are populated in the network interface receive ring. When the CPU receives network packets from the network interface, the CPU allocates a free buffer from the preallocated pool and replenishes the receive ring. When the CPU has finished packet processing, the CPU adds the memory buffer associated with this network packet to the free buffer pool. Packets read from the buffers are always the same size as those written, so no explicit zeroing or overwriting of buffers on allocation is required.

Each pre-allocated memory buffer is fixed at 1792 bytes, and is sufficient to hold a standard Ethernet frame. When a frame is received by the network interface, a proprietary header is prepended onto the frame indicating its length, among other parameters. The frame is then sent to the CPU. The CPU reads the length out of the proprietary header and uses this to process the frame out of the buffer. For standard Ethernet frames, only a single frame is placed into a memory buffer – buffers do not contain multiple frames.

Jumbo packets are supported – these must be split across multiple memory buffers. The proprietary header ensures that the packet is reconstructed correctly.

The length in the proprietary header is always correct – the product would not function if this were not the case.

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_RIP.2: Packets read from the buffers are always the same size as those written, so no explicit zeroing or overwriting of buffers on allocation is required (i.e., packet content is always overwritten before being read).

## 6.4 Identification and authentication

The TOE supports role-based authentication. Wireless users (or clients, term used interchangeably) can authenticate to an external authentication server or to the Controller's internal database. The administrator can create a user

account in the internal database and assign a predefined role to that account. User log in to the Controller are restricted based on their assigned role. In this case, the authentication mechanism is provided by the TOE and the credentials are maintained in the internal database. The administrator can also configure the TOE so that wireless users are authenticated using an external authentication server[9]. The TOE supports the RADIUS, LDAP, and TACACS+ servers. A trusted channel is established between the TOE and the authentication server. For wireless users using 802.1X authentication, when a user client connects to the TOE, the TOE passes authentication protocol messages between the client and the authentication server, until the user is authenticated, or authentication is denied. As a part of the initial handshake, the authentication server presents to the client a TLS server certificate. Communications between the client and the server are then encrypted by AES. The following authentication protocols are supported: EAP-TLS, EAP-TTLS, PEAP.

For EAP-TTLS and PEAP protocols, the user will authenticate to the server over a TLS encrypted connection using a username and password. For EAP-TLS, the user will use a X.509 client certificate[10] to authenticate. The certificate will contain the username of the user, and may contain other user-specific information. The authentication server will maintain a list of trusted certification authorities to verify the client certificate. If the authentication fails, the authentication server will communicate the authentication failure to the TOE. Otherwise, the authentication server will communicate the authentication success to the TOE and send to the TOE the session key, which was derived during the EAP-TLS/EAP-TTLS/PEAP handshake, as well as the user role attribute. The session key may be used by the TOE to encrypt further communications with a wireless client.

For users connecting with a VPN client, the IPsec/IKE VPN is established between the TOE and the client prior to the user authentication using pre-shared keys or certificates, and can optionally authenticate to the external authentication server using a username and password. The external authentication server communicates success or failure of the authentication to the TOE.

The TOE accepts pre-shared keys for IPsec (IKEv1, IKEv2) and WPA2 (WPA-PSK, aka WPA-Personal). It accepts bit-based pre-shared keys for all of these protocols, and it accepts text-based PSKs that are transformed into bit-based PSKs only for WPA2. Text-based keys are conditioned using PBKDF2, as specified in 802.11i.

When a wireless user exceeds the configured authentication threshold, the user is automatically blacklisted by the controller, an event is logged, and an SNMP trap is sent (optional SNMP server in the operating environment must be set up to capture SNMP traps). By default, the maximum authentication failure threshold is set to 0 (but can be set as high as 255), which means that there is no limit to the number of times a user can attempt to authenticate. When users are blacklisted because they exceed the authentication failure threshold, they are blacklisted indefinitely by default. Administrator can configure the duration of the blacklisting. Please refer to the “Management Password Policy” section in the Aruba OS User Guide documentation for information on configuring blacklisting.

Remote administrators are configured as users who have privileges to access the CLI and Web GUI administration interfaces. Remote administrators are authenticated as users using a local database or external authentication server. A trusted channel is established between the TOE and the authentication server. The remote administrators authenticate as users using a username and password via Web GUI and username/password for SSH. The Web GUI interface provides a trusted path to connect to the TOE via HTTPS. The HTTPS interface uses a server RSA or ECDSA certificate which is stored on the TOE.

The controller maintains a counter of failed authentication attempts for a given administrative username within the past three minutes. An unsuccessful authentication attempt is detected when an invalid password is entered for a valid username. If the failed authentication threshold is reached for a given username, that user account is locked out for the configured lock-out period. Note that no indication is given to the remote system attempting to log in that the account has been locked out – authentication simply fails as though an incorrect password were provided. Additionally, no indication is given to the remote system whether a given username was valid or not valid.

In general, Aruba uses an automated test tool (Ixia IxANVL) to test conformance with RFCs and 802.1X. Information about IxANVL is at http://www.ixiacom.com/products/network_test/applications/ixanvl/. Aruba also uses VeriWave test tools (http://www.ixiacom.com/solutions/wifi-performance-test/) to test Wi-Fi performance – one component of the VeriWave test suite is to exercise 802.1X capabilities of the product. Aruba also conducts

[9] The Controller accepts the user credentials and sends the credentials to the authentication server. The wireless clients never communicate directly with the authentication server.
[10] When authentication server is used, the credentials (password or certificate) are maintained outside the TOE.

interoperability testing through custom-built automated test beds which contain numerous client operating systems (Windows XP, Windows Vista, Windows 7, Windows 8, Mac OS X, Linux, Apple iOS, Android, etc.) connecting to Aruba Wi-Fi access points. Finally, the products are Wi-Fi certified by the Wi-Fi Alliance – conformance with 802.1X and EAP/RADIUS are requirements to pass these tests.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_8021X_EXT.1: see above
- FIA_AFL.1: After an administrator specified numbers of failed attempts, the TOE will lockout (blacklist) the offending remote administrator, log the event, and send a SNMP trap. The offending administrator will remain locked out until the lock-out period has expired. The administrator can configure the lock-out period:
    - password-lock-out Configuring the number of failed attempts within 3 minute window to lockout the user. Provides ability to reduce the number of passwords that can be guessed in a short time. Automatically clears the lockout after configured "lock-out" minutes. Range: 0-10 attempts. Default: 0 (lockout of users is disabled by default).
    - password-lock-out-time Configuring the number of minutes the user is locked out. The lockout is cleared without administrator intervention. Range: 1 min to 1440 min (24 hrs). Default: 3 min.
        - FIA_AFL.1 is enforced by the TOE and when using external authentication server.
- FIA_PMG_EXT.1: The TOE authentication mechanism provides configuration for minimum password length. The administrator should at a minimum, requires password to be at least 6 characters long. The following calculation is based on the following facts:
    - Password is case-sensitive
    - A-Z, a-z, 0-9, !@#$%^&*()_+, and extended characters
    - Password minimum length is set to 8
    - Passwords have maximum lifetime and new passwords must contain a minimum of 4 character changes from the previous password

  Passwords must be at least eight characters long. Numeric, alphabetic (upper and lower case), and keyboard/extended characters can be used, which gives a total of 95 characters to choose from. An eight character password using all characters has 95^8 total possible combinations. The probability for a random attempt to succeed is therefore less than one in 1,000,000,000,000,000.
- FIA_PSK_EXT.1: The TOE accepts between 22-64 character text based pre-shared keys ( composed of any combination of upper and lower case letters, numbers, and special characters (that include: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')')) for IPsec, WPA2 and IKEv1/IKEv2.
- FIA_UAU.6 requires a user to reauthenticate when a password is changed or the session is locked
- FIA_UAU.7: The TOE provides only obscured feedback to the administrative user while authentication is in progress at the local console by displaying an asterisk (*) for each character entered.
- FIA_UAU_EXT.5: The TOE provides local accounts and can also be configured to utilize LDAP, RADIUS, and TACACS+ authentication servers in its operational environment. The administrator can configure TOE to provide the same or different authentication mechanism (local, remote) for wireless users and administrators. The TOE shall invoke the correct authentication mechanism as configured by the administrator.
- FIA_UIA_EXT.1: Prior to establishing an administrative user session the TSF displays an Authorized Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE (FTA_TAB.1). The TOE requires an administrator to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_X509_EXT.1: The TOE protects, stores and allows authorized administrators to load X.509v3 certificates for use to support authentication for IPsec, TLS and SSH connections. Certificates are loaded

into the controller using the "Certificate Manager" section of the Web-based user interface. The controller supports loading of certificates in PEM, DER, or PFX format. Private keys may be loaded onto the controller through a password-protected PFX file, or may originate on the controller at the time a Certificate Signing Request (CSR) is generated.

- During runtime, certificates and private keys are stored in ramdisk, in volatile memory, in decrypted form. This allows private keys to be accessed rapidly for high network load conditions. When powered off, private keys are stored encrypted in non-volatile (flash) memory. The encryption method used is AES256 using the Private Key Encrypting Key (PKEK) described in the CSP table. The PKEK is ultimately protected by hardware (TPM).

## 6.5 Security management

The TOE provides the administrator role the capability to enable the management of security attributes, TSF data and security functions. The administrator can configure TOE security settings and policies using the Web Graphical User Interface via HTTPS, or the command line interface via serial console locally or remotely using SSH. The Web GUI is a just a front-end to the CLI (i.e., calls the CLI internally). It is provided as a user-friendly interface for the administrator to manage the TOE. Every function that can be performed on the Web GUI, can also be performed using the CLI but not vice versa. However, every security management function claimed can be done either using the Web GUI or CLI.

The TOE supports role-based authentication. There are three types of roles: administrator[11] role, limited administrator role, and wireless user role. The limited administrator role and wireless user role are not TOE Security management roles. Administrators can manage the TOE using HTTPS Web GUI or CLI. Wireless clients cannot access the TOE through the Web GUI or CLI interfaces and, therefore, do not have access to the management functionalities of the TOE. The limited administrator role can perform non-security tasks which include the following:

- Read-only role - This role permits access to CLI "show" commands or Web GUI monitoring pages only. It does not allow user to perform any action such as copying files or rebooting the controller.
- Network operations role - This role permits access to Monitoring, Reports, and Events pages in the Web GUI that are useful for monitoring the controller. This role can log into the CLI; however, user can only use a subset of CLI commands to monitor the controller.

The remote administrator or limited administrator authenticates with a username and password or certificate via an HTTPS connection or via the interactive command line. Once the administrator is authenticated, the TOE provides management interfaces which can be used by the administrator to configure the TOE security functions. Local administrators can also use the CLI via a serial console (direct) connection to the TOE by using username and password. Remote administrators may use the Web GUI interface from the browser or may also use the CLI interface via an SSH protocol connection from an SSH client. An administrator or limited administrator can log on to the administrative interface (Web GUI or SSH) while connected to the network over wireless as long as appropriate firewall policies have been configured.

The TOE provides to the administrator capabilities manage all security functions identified in this Security Target, including configuring banner warnings and idle timeout limits. For more information about the management interfaces, please refer to the ArubaOS User Guide documentation. To find out more information about the cryptographic functionalities, please refer to the FIPS 140-2 Security Policies.

Note that there are no services available that do not require authentication. Furthermore, administrators can initiate secure TOE updates in accordance with FPT_TUD_EXT.1.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1: Only authorized administrators in the administrator role can enable, disable, determine and modify the behavior of the TOE security functions.
- FMT_MTD.1(1): Only authorized administrators in the administrator role can manage TSF data.

[11] Some Aruba documents may refer to as the "root" and/or "crypto officer" role.

- FMT_MTD.1(2): The TOE provides no interfaces that allow user passwords to be read. Passwords are not stored in plaintext. They are stored on flash using a SHA1 hash. They are used for administrative authentication. Wireless user passwords are found in an external AAA/RADIUS server.
- FMT_MTD.1(3): The TOE provides no interfaces that allow pre-shared, symmetric, or private keys to be read.
- FMT_SMF.1: The TOE provides administrative functions to manage all TOE security functions identified in this Security Target including but not limited to management of cryptographic functions, secure TOE updates, banner message configuration, etc.
- FMT_SMR.1: The TOE supports role-based authentication. Administrators can make use of both local and remotely accessible administrator interfaces.

## 6.6 Protection of the TSF

Communication between the Mobility Controller and APs is protected in two manners. All control traffic (i.e., TSF data) is protected from disclosure and modification using IPsec. All data traffic is protected in the same way that it is protected over-the-air since that traffic is encrypted and decrypted at the Mobility Controller and not at the APs.

The Mobility Controller has an internal battery-backed hardware clock that provides reliable time stamps used for auditing. The internal clock may be synchronized (not required) with a time signal obtained from an external NTP server.

The Mobility Controller and AP run a suite of self-tests during power-up and periodically during operation, which includes demonstration of the correct operation of the hardware and the use of cryptographic functions to verify the integrity of TSF executable code and static data. An administrator can choose to reboot the TOE to perform power-up self-tests. The Mobility Controller and AP runs the suite of FIPS 140-2 validated cryptographic module self-tests during start-up or on request from the administrator, including immediately after generation of a key (FIPS self-tests, including the continuous RNG test). If a self-test fails, the TOE will immediately halt operation thereby preventing potentially insecure operations (i.e., maintaining a secure state). The controller will reboot after a self-test failure. During reboot, memory is re-initialized, which wipes all keys and user data. If a self-test failure continues to occur, the controller will continue to reboot repeatedly.

The TOE uses IPsec, TLS, and SSH for all communications terminating at the TOE. Each of these protocols includes features to detect replayed data and the TOE will reject any such data that is received.

In order to update the firmware, the administrator can download the firmware file from the Aruba support website. Using TFTP, FTP, SCP, or HTTPS (Web UI only), the administrator can import the image into the controller.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_FLS.1: The TOE will stop when power-on self-tests fail in order to prevent entering an insecure operational state.
- FPT_ITT.1: The TOE protects TSF data sent between the Mobility Controller and APs using IPsec.
- FPT_RPL.1: The TOE can detected repeated IPsec, TLS, or SSH packets and will discard any such packets when received.
- FPT_STM.1: The TOE provides its own time and/or relies on an external trusted time server for this function.
- FPT_TST_EXT.1: The TOE offers a suite of self-tests for administrator (in FIPS terminology, the Crypto Officer which is mapped to the administrator role) to verify the correct operation of the key generation and static TSF cryptographic data. Software images are hashed and cryptographically signed, and an image with an invalid signature will not be copied by the controller into the image partition. Likewise, a software image stored in the image partition through external means will be rejected by the hardware bootloader if the image hash or signature is invalid. RSA2048 and SHA1 are used to sign the image. The signing certificate is issued by Aruba's internal CA. This CA is stored offline with private keys protected by an HSM. The public root CA certificate is programmed into the bootROM of all Aruba products at the time of manufacturing. On the Aruba 7200 series controller, the public root CA certificate is programmed into the

CPU and electronic fuses are blown to protect it from overwrite. On bootup, the controller performs a SHA1 hash of the ArubaOS image file, then compares it to the digital signature. The digital signature is checked against the root CA certificate. Note that since the AP firmware images are packaged within the controller image file, the APs are automatically upgraded with the controller upgrade, since the APs check for version mismatches between the controller and themselves.

- FPT_TUD_EXT.1: The TOE allows administrators to query the current version of its firmware/software and allows those administrators to initiate firmware/software updates. Prior to installing any update the administrator can verify the digital signature of the update or alternately ensure that the software hash of the update matches a value published by Aruba. If in the case there is unsuccessful verification of candidate updates, the controller will reboot (and continue to reboot until an administrator intervenes) if this happens during bootup. If the verification fails at the time an image is loaded, the controller will indicate an error message and will not write the image file to flash.

## 6.7 Resource utilization

The TOE includes rate limiting for traffic from untrusted users reaching the control plane. By default, an untrusted user may send no more than 100 packets per second to the control function of the TOE. This rate is configurable from 1 to 255. The system may be configured to blacklist users who exceed the threshold. Once blacklisted, all network traffic from the user is rejected until the user is removed from the blacklist (either through expiration of a timer or through administrator action.) The blacklist action will be logged in the audit log, in compliance with FAU_GEN.1.2.

Other features are present in the TOE to protect the control plane. These include protection against invalid IP addresses in the user table, use of a control-plane firewall to control network traffic that is permitted to reach the control plane, and aggregate bandwidth limitations on traffic reaching the control plane. These features do not generate audit messages, however, and so do not meet the requirements of FAU_GEN.1.2.

The Resource utilizsation function is designed to satisfy the following security functional requirements:

- FRU_RSA.1: The TOE enforces the maximum quota on the amount of control-plane bandwidth that can be consumed by an untrusted user.

## 6.8 TOE access

Whether connecting to the CLI (remotely) and web GUI, the TOE displays an advisory message when an administrator logs on. The message is configurable by TOE administrators.

The TOE terminates a wireless user session or an administrator CLI session (for a local or remote administrator) after session inactivity time exceeds a configurable session idle timeout. The session idle timeout is the maximum amount of time a wireless user or an administrator may remain idle.

However, the timeout does not apply to the following web GUI screens below which are auto refreshed. These screens are used for monitoring purposes only and do not provide any security management interface. The information on these screens are updated constantly; therefore, the screens are never idle (timeout does not apply).

- *Monitoring > Network > All Access Points*
- *Monitoring > Network > All Air Monitor*
- *Monitoring > Network > Wired Access Points*
- *Monitoring > Network > All WLAN Clients*
- *Monitoring > Controller > Access Points*
- *Monitoring > Controller > Air Monitor*
- *Monitoring > Controller > Wired Access Points*

- *Monitoring > Controller > Clients*
- *Monitoring > WLAN > [ESSID_NAME] > Access Points*
- *Monitoring > WLAN > [ESSID_NAME] > Clients*
- *Monitoring > Debug > Local Clients*
- *Monitoring > Debug > Process Logs*
- *Maintenance > WLAN > Program AP*
- *Maintenance > WLAN > Reboot AP*

The TOE assesses wireless user inactivity as the cessation of network traffic arriving from the wireless client. It should be noted that processes acting on behalf of the user may send protocol network packets to the mobility controller, even when the user is not interacting directly, e.g. pressing keys.

To change the session idle timeout, the administrator can use the "loginsession timeout" command of the CLI. The "web-server session-timeout" command applies to the WebUI. The default value of the session idle timeout is 15 minutes.

Of course, administrators can terminate their own sessions at any time simply by using the CLI or web GUI function to log off. Wireless users can also disconnect at any time, terminating their session.

In order to limit access to the administrative functions, the TOE can be configured to deny wireless clients and administrative sessions based on the time/date, IP address (location), as well as information retained in a blacklist. Firewall rules are used to restrict access, and can be configured to blacklist clients when a rule is violated. Unlike the other properties, the blacklist is dynamically managed by the TOE identifying potentially undesirable network devices based on observed activities. If a device is actively identified in the blacklist, it cannot be used to connect to an administrative interface.

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_SSL.3: By default, the TOE will terminate inactive user session after 15 minutes and require users to login again. The timeout period can be changed only by administrator.
- FTA_SSL.4: Administrative users can log off at any time by issuing the applicable command.
- FTA_SSL_EXT.1: Local inactive administrator sessions are terminated, just like remote inactive administrator sessions, after the configured timeout period.
- FTA_TAB.1: The TOE displays an advisory warning banner regarding use of the TOE prior to establishing an administrator session. The administrator can configure the warning message displayed in the banner.
- FTA_TSE.1: The TOE can deny establishment of a wireless client session based on location, time, day, and blacklist state.

## 6.9 Trusted path/channels

The TOE provides trusted paths for remote administration and trusted channels for communication between itself and peers in the operating environment including authentication, audit, and NTP servers.

For remote administrators, the TOE uses HTTPS/TLS to offer secure remote web GUI-based administration and SSH to offer a secure remote administration CLI.

For wireless users using an open system connection, the TOE provides an IPsec/IKE VPN trusted path from the TOE to the wireless users for authentication of the wireless users. A pre-shared key or certificate is distributed using an out-of-band method and is the basis for initial authentication. The user then optionally authenticates to the external authentication server using a username and password.

For wireless users operating in a Robust Security Network (RSN), IEEE 802.11-2007 is used to provide a trusted channel between the TOE and wireless clients.

The TOE uses the IPsec/IKE protocol with pre-shared keys or certificates to establish a trusted channel between itself and the external authentication, logging, and NTP servers. To configure the channels the administrator uses the Security -> Advanced -> VPN panel of the Web GUI to create the host-to-host IPsec/IKE connections. All configuration settings must specify FIPS-certified encryption algorithms as specified by the FCP_COP.1 requirements.

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1: The TOE uses the IPsec/IKE protocol with pre-shared keys or certificates to establish a trusted channel between itself and external authentication, logging, and NTP servers.
- FTP_TRP.1: The TSF uses SSH, TLS/HTTPS to provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.

# 7. Protection Profile Claims

The ST conforms to the *Protection Profile for Wireless Local Area Network (WLAN) Access Systems*, version 1.0, 01 December 2011 (WLASPP). As explained previously, the security problem definition, security objectives, and security requirements have been drawn verbatim from the WLASPP.

# 8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Requirement Dependencies;
- TOE Summary Specification.

## 8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

### 8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

| | P.ACCESS_BANNER | P.ACCOUNTABILITY | P.ADMIN_ACCESS | P.COMPATIBILITY | P.EXTERNAL_SERVERS | T.ADMIN_ERROR | T.RESOURCE_EXHAUSTION | T.TSF_FAILURE | T.UNAUTHORIZED_ACCESS | T.UNAUTHORIZED_UPDATE | T.UNDETECTED_ACTIONS | T.USER_DATA_REUSE | A.NO_GENERAL_PURPOSE | A.NO_TOE_BYPASS | A.PHYSICAL | A.TRUSTED_ADMIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **O.AUTH_COMM** | | | | | | | | | X | | | | | | | |
| **O.CRYPTOGRAPHIC_FUNCTIONS** | | | X | | | | | | X | | | | | | | |
| **O.DISPLAY_BANNER** | X | | | | | | | | | | | | | | | |
| **O.FAIL_SECURE** | | | | | | | | X | | | | | | | | |
| **O.PROTECTED_COMMUNICATIONS** | | | X | | | | | | X | | | | | | | |
| **O.PROTOCOLS** | | | | X | X | | | | | | | | | | | |
| **O.REPLAY_DETECTION** | | | | | | | | | X | | | | | | | |
| **O.RESIDUAL_INFORMATION_CLEARING** | | | | | | | | | | | | X | | | | |
| **O.RESOURCE_AVAILABILITY** | | | | | | | X | | | | | | | | | |
| **O.ROBUST_TOE_ACCESS** | | X | | | | | | | X | | | | | | | |
| **O.SESSION_LOCK** | | | | | | | | | X | | | | | | | |
| **O.SYSTEM_MONITORING** | | X | | | | | | | | | X | | | | | |
| **O.TIME_STAMPS** | | X | | | | | | | | | | | | | | |
| **O.TOE_ADMINISTRATION** | | | X | | | X | | | X | | | | | | | |
| **O.TSF_SELF_TEST** | | | | | | | | X | | | | | | | | |
| **O.VERIFIABLE_UPDATES** | | | | | | | | | | X | | | | | | |
| **O.WIRELESS_CLIENT_ACCESS** | | | | | | | | | X | | | | | | | |
| **OE.NO_GENERAL_PURPOSE** | | | | | | | | | | | | | X | | | |
| **OE.NO_TOE_BYPASS** | | | | | | | | | | | | | | X | | |
| **OE.PHYSICAL** | | | | | | | | | | | | | | | X | |
| **OE.TRUSTED_ADMIN** | | | | | | X | | | | | | | | | | X |

**Table 7 Environment to Objective Correspondence**

#### 8.1.1.1 P.ACCESS_BANNER

*The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.*

This Organizational Policy is satisfied by ensuring that:

- O.DISPLAY_BANNER: Satisfies this policy by ensuring that the TOE displays an Authorized Administrator configurable banner that provides all users with a warning about the unauthorized use of the TOE.

#### 8.1.1.2 P.ACCOUNTABILITY

*The authorized users of the TOE shall be held accountable for their actions within the TOE.*

This Organizational Policy is satisfied by ensuring that:

- O.ROBUST_TOE_ACCESS: Supports this policy by requiring the TOE to identify and authenticate all administrators prior to allowing any TOE access or any TOE mediated access on behalf of those administrators.
- O.SYSTEM_MONITORING: Supports this policy by providing the administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user.
- O.TIME_STAMPS: Plays a role in supporting this policy by requiring the TOE to provide a reliable time stamp. This will be used when audit records are generated, allowing administrators to tie auditable actions to the time those actions took place, perhaps on disparate systems. This ability aids in proving accountability for users whose actions cause those audit records to be generated.

#### 8.1.1.3 P.ADMIN_ACCESS

*Administrators shall be able to administer the TOE both locally and remotely through protected communications channels.*

This Organizational Policy is satisfied by ensuring that:

- O.CRYPTOGRAPHIC_FUNCTIONS: Contributes to mitigating this threat by providing the underlying cryptographic functionality required by other protection mechanisms.
- O.PROTECTED_COMMUNICATIONS: Contributes to mitigating this threat by ensuring protection of the communication between the TOE and authorized administrator while transmitting data.
- O.TOE_ADMINISTRATION: Supports this policy by requiring the TOE to provide mechanisms (e.g., local authentication, remote authentication, means to configure and manage the TOE both remotely and locally) that allow remote and local administration of the TOE.

#### 8.1.1.4 P.COMPATIBILITY

*The TOE must meet Request for Comments (RFC) requirements for implemented protocols to facilitate inter-operation with other network equipment (e.g., certificate authority, NTP server) using the same protocols.*

This Organizational Policy is satisfied by ensuring that:

- O.PROTOCOLS: Satisfies this policy by requiring that standardized protocols are implemented in the TOE to ensure interoperability among IT entities using the same protocols.

#### 8.1.1.5 P.EXTERNAL_SERVERS

*The TOE must support standardized (RFCs) protocols for communication with a centralized audit server and a RADIUS authentication server.*

This Organizational Policy is satisfied by ensuring that:

- O.PROTOCOLS: Satisfies the policy by ensuring that the TOE can communicate with an external audit server and RADIUS authentication server, even when auditing and authentication are also provided locally.

#### 8.1.1.6 T.ADMIN_ERROR

*An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.*

This Threat is satisfied by ensuring that:

- O.TOE_ADMINISTRATION: Plays a role in mitigating this threat by limiting the functions an administrator can perform. Revoking administrator access when not needed also reduces the chance that an error may occur.
- OE.TRUSTED_ADMIN: Mitigates this threat by ensuring the administrators are properly trained and the administrative guidance instructs the administrator how to properly configure the environment and TOE to avoid mistakes.

#### 8.1.1.7 T.RESOURCE_EXHAUSTION

*A process or user may deny access to TOE services by exhausting critical resources on the TOE.*

This Threat is satisfied by ensuring that:

- O.RESOURCE_AVAILABILITY: Mitigates the threat by ensuring that the TOE has mechanisms and policy in place to deal with attempts to exhaust resources.

#### 8.1.1.8 T.TSF_FAILURE

*Security mechanisms of the TOE may fail, leading to a compromise of the TSF.*

This Threat is satisfied by ensuring that:

- O.FAIL_SECURE: Contributes to mitigating this threat by ensuring that on a detected failure the TOE maintains a secure state.
- O.TSF_SELF_TEST: Counters this threat by ensuring that the TSF runs a suite of self tests to successfully demonstrate the correct operation of the TSF.

#### 8.1.1.9 T.UNAUTHORIZED_ACCESS

*A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.*

This Threat is satisfied by ensuring that:

- O.AUTH_COMM: Works to mitigate this threat by ensuring that the TOE identifies and authenticates all users prior to allowing TOE access or setting up a security association with that user. The TOE must also be capable of sending its own credentials to users to ensure mutual authentication prior to obtain identification and authentication data.
- O.CRYPTOGRAPHIC_FUNCTIONS: Contributes to mitigating this threat by providing the underlying cryptographic functionality required by other protection mechanisms.
- O.PROTECTED_COMMUNICATIONS: Contributes to mitigating this threat by ensuring protection of the communication between the TOE and authorized administrator while transmitting data.
- O.REPLAY_DETECTION: Prevents unauthorized access by replaying sessions (or portions of sessions) from legitimate administrators or entities that have been captured by a malicious actor.
- O.ROBUST_TOE_ACCESS: Mitigates this threat by requiring the TOE to identify and authenticate all administrators prior to allowing any TOE access or any TOE mediated access on behalf of those administrators.
- O.SESSION_LOCK: Mitigates this threat by requiring the TOE to provide a way for the user to lock a session or for the TOE to lock after a certain time-period which ensures an authorized session cannot be hijacked at the terminal.

- O.TOE_ADMINISTRATION: Requires the TOE to provide mechanisms (e.g., local authentication, remote authentication, means to configure and manage the TOE both remotely and locally) that allow remote and local administration of the TOE.
- O.WIRELESS_CLIENT_ACCESS: Mitigates the threat by providing mechanisms to restrict wireless client access according to the desired security posture of the TOE.

#### 8.1.1.10 T.UNAUTHORIZED_UPDATE

*A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.*

This Threat is satisfied by ensuring that:

- O.VERIFIABLE_UPDATES: Ensures that the administrator can confirm the update.

#### 8.1.1.11 T.UNDETECTED_ACTIONS

*Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.*

This Threat is satisfied by ensuring that:

- O.SYSTEM_MONITORING: Mitigates this threat by providing the administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user.

#### 8.1.1.12 T.USER_DATA_REUSE

*User data may be inadvertently sent to a destination not intended by the original sender.*

This Threat is satisfied by ensuring that:

- O.RESIDUAL_INFORMATION_CLEARING: Counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.

#### 8.1.1.13 A.NO_GENERAL_PURPOSE

*It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of the TOE.*

This Assumption is satisfied by ensuring that:

- OE.NO_GENERAL_PURPOSE: Ensures the TOE does not include any general-purpose computing or storage capabilities. This will protect the TSF data from malicious processes.

#### 8.1.1.14 A.NO_TOE_BYPASS

*Information cannot flow between the wireless client and the internal wired network without passing through the TOE.*

This Assumption is satisfied by ensuring that:

- OE.NO_TOE_BYPASS: Ensures that all information flow between external and internal networks in different enclaves passes through the TOE.

#### 8.1.1.15 A.PHYSICAL

*Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.*

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: Ensures the TOE, the TSF data, and protected user data is protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized

intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment.

#### 8.1.1.16 A.TRUSTED_ADMIN

*TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.*

This Assumption is satisfied by ensuring that:

- OE.TRUSTED_ADMIN: Ensures the administrators are properly trained and the administrative guidance instructs the administrator how to properly configure the environment and TOE to avoid mistakes.

## 8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 8** indicates the requirements that effectively satisfy the individual objectives. .

### 8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

| | O.AUTH_COMM | O.CRYPTOGRAPHIC_FUNCTIONS | O.DISPLAY_BANNER | O.FAIL_SECURE | O.PROTECTED_COMMUNICATIONS | O.PROTOCOLS | O.REPLAY_DETECTION | O.RESIDUAL_INFORMATION_CLEARING | O.RESOURCE_AVAILABILITY | O.ROBUST_TOE_ACCESS | O.SESSION_LOCK | O.SYSTEM_MONITORING | O.TIME_STAMPS | O.TOE_ADMINISTRATION | O.TSF_SELF_TEST | O.VERIFIABLE_UPDATES | O.WIRELESS_CLIENT_ACCESS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **FAU_GEN.1** | | | | | | | | | | | | X | | | | | |
| **FAU_GEN.2** | | | | | | | | | | | | X | | | | | |
| **FAU_SAR.1** | | | | | | | | | | | | X | | | | | |
| **FAU_SAR.2** | | | | | | | | | | | | X | | | | | |
| **FAU_SEL.1** | | | | | | | | | | | | X | | | | | |
| **FAU_STG.1** | | | | | | | | | | | | X | | | | | |
| **FAU_STG_EXT.1** | | | | | X | | | | | | | X | | | | | |
| **FAU_STG_EXT.3** | | | | | | | | | | | | X | | | | | |
| **FCS_CKM.1(1)** | | X | | | | | | | | | | | | | | | |
| **FCS_CKM.1(2)** | | X | | | | | | | | | | | | | | | |
| **FCS_CKM.2(1)** | | X | | | | | | | | | | | | | | | |
| **FCS_CKM.2(2)** | | X | | | | | | | | | | | | | | | |
| **FCS_CKM_EXT.4** | | X | | | | | | X | | | | | | | | | |
| **FCS_COP.1(1)** | | X | | | | | | | | | | | | | | | |
| **FCS_COP.1(2)** | | X | | | | | | | | | | | | | | X | |
| **FCS_COP.1(3)** | | X | | | | | | | | | | | | | | X | |
| **FCS_COP.1(4)** | | X | | | | | | | | | | | | | | | |
| **FCS_COP.1(5)** | | X | | | | | | | | | | | | | | | |
| **FCS_HTTPS_EXT.1** | X | | | | X | X | | | | | | | | | | | |
| **FCS_IPSEC_EXT.1** | X | | | | X | X | | | | | | | | | | | |
| **FCS_RBG_EXT.1** | | X | | | | | | | | | | | | | | | |
| **FCS_SSH_EXT.1** | X | | | | X | X | | | | | | | | | | | |
| **FCS_TLS_EXT.1** | X | | | | X | X | | | | | | | | | | | |
| **FDP_RIP.2** | | | | | | | | X | | | | | | | | | |

| | O.AUTH_COMM | O.CRYPTOGRAPHIC_FUNCTIONS | O.DISPLAY_BANNER | O.FAIL_SECURE | O.PROTECTED_COMMUNICATIONS | O.PROTOCOLS | O.REPLAY_DETECTION | O.RESIDUAL_INFORMATION_CLEARING | O.RESOURCE_AVAILABILITY | O.ROBUST_TOE_ACCESS | O.SESSION_LOCK | O.SYSTEM_MONITORING | O.TIME_STAMPS | O.TOE_ADMINISTRATION | O.TSF_SELF_TEST | O.VERIFIABLE_UPDATES | O.WIRELESS_CLIENT_ACCESS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **FIA_8021X_EXT.1** | X | | | | X | X | | | | | | | | | | | |
| **FIA_AFL.1** | | | | | | | | | | X | | | | | | | |
| **FIA_PMG_EXT.1** | | | | | | | | | | X | | | | X | | | |
| **FIA_PSK_EXT.1** | X | | | | | | | | | | | | | | | | |
| **FIA_UAU.6** | | | | | | | | | | X | | | | | | | |
| **FIA_UAU.7** | | | | | | | | | | X | | | | | | | |
| **FIA_UAU_EXT.5** | | | | | | | | | | X | | | | X | | | |
| **FIA_UIA_EXT.1** | X | | | | | | | | | X | | | | | | | |
| **FIA_X509_EXT.1** | | X | | | | | | | | | | | | | | | |
| **FMT_MOF.1** | | | | | | | | | | | | | | X | | | |
| **FMT_MTD.1(1)** | | | | | | | | | | | | | | X | | | |
| **FMT_MTD.1(2)** | | | | | | | | | | | | | | X | | | |
| **FMT_MTD.1(3)** | | | | | | | | | | | | | | X | | | |
| **FMT_SMF.1** | | | | | | | | | | | | | | X | | | |
| **FMT_SMR.1** | | | | | | | | | | X | | | | X | | | |
| **FPT_FLS.1** | | | | X | | | | | | | | | | | X | | |
| **FPT_ITT.1** | | | | | X | | | | | | | | | | | | |
| **FPT_RPL.1** | | | | | X | | X | | | | | | | | | | |
| **FPT_STM.1** | | | | | | | | | | | | X | X | | | | |
| **FPT_TST_EXT.1** | | | | | | | | | | | | | | | X | | |
| **FPT_TUD_EXT.1** | | | | | | | | | | | | | | | | X | |
| **FRU_RSA.1** | | | | | | | | | X | | | | | | | | |
| **FTA_SSL.3** | | | | | | | | | | X | X | | | | | | |
| **FTA_SSL.4** | | | | | | | | | | X | X | | | | | | |
| **FTA_SSL_EXT.1** | | | | | | | | | | X | X | | | | | | |
| **FTA_TAB.1** | | | X | | | | | | | | | | | | | | |
| **FTA_TSE.1** | | | | | | | | | | | | | | | | | X |
| **FTP_ITC.1** | X | | | | X | X | | | | | | | | | | | |
| **FTP_TRP.1** | X | | | | X | | | | | | | | | X | | | |

**Table 8 Objective to Requirement Correspondence**

#### 8.2.1.1 O.AUTH_COMM

*The TOE will provide a means to ensure users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity.*

This TOE Security Objective is satisfied by ensuring that:

- FIA_X8021X_EXT.1 provides the two-way authentication necessary to allow a wireless client access to the wired network, and serves as a part of the 802.11-2007 WPA2 protocol to establish the communication channel with the wireless client.
- FCS_HTTPS_EXT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.

- FCS_IPSEC_EXT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FCS_SSH_EXT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FCS_TLS_EXT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FIA_PSK_EXT.1: Requires the TOE support the formation of strong pre-shared keys (either though a large character set for text-based pre-shared keys, or through generation by the TOE's (or an off-box) RBG function) that can be used to mutually authenticate the TOE and its communication partner.
- FIA_UIA_EXT.1: Requires administrators (including remote administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path.
- FTP_ITC.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FTP_TRP.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.

#### 8.2.1.2 O.CRYPTOGRAPHIC_FUNCTIONS

*The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE, or stored outside the TOE.*

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1(1): Generates symmetric and asymmetric key, respectively. These keys are used by the AES encryption/decryption functionality specified in FCS_COP.1(5) and used for cryptographic signatures as specified in FCS_COP.1(2).
- FCS_CKM.1(2): Generates symmetric and asymmetric key, respectively. These keys are used by the AES encryption/decryption functionality specified in FCS_COP.1(5) and used for cryptographic signatures as specified in FCS_COP.1(2).
- FCS_CKM.2(1): Assures that the distribution method of cryptographic keys for wireless client communications are in accordance with a standard and do not get exposed.
- FCS_CKM.2(2): Assures that the distribution method of cryptographic keys for wireless client communications are in accordance with a standard and do not get exposed.
- FCS_CKM_EXT.4: Provides the functionality for ensuring key and key material is zeroized. This applies not only to key that resides in the TOE, but also to intermediate areas (physical memory, page files, memory dumps, etc.) where key material may appear.
- FCS_COP.1(1): Specifies that AES be used to perform encryption and decryption operations for the various protocols specified in the PP.
- FCS_COP.1(2): Requires a digital signature capability be implemented in the TOE for trusted updates and certificate operations associated with identification and authentication of authorized IT entities and remote administrators.
- FCS_COP.1(3): Requires that the TSF provide hashing services using an implementation of the Secure Hash Algorithm algorithms for data integrity verification and non-data integrity operations.
- FCS_COP.1(4): Requires that the TSF provide hashing services using an implementation of the Secure Hash Algorithm algorithms for data integrity verification and non-data integrity operations.
- FCS_COP.1(5): Specifies that AES be used to perform encryption and decryption operations for the various protocols specified in the PP.
- FCS_RBG_EXT.1: Ensures that keying material is robustly generated.
- FIA_X509_EXT.1: Requires that the certificates used to support many of the cryptographic operations previously mentioned conform to an appropriate standard.

#### 8.2.1.3 O.DISPLAY_BANNER

*The TOE will display an advisory warning regarding use of the TOE.*

This TOE Security Objective is satisfied by ensuring that:

- FTA_TAB.1: Requires the TOE to display an administrator defined banner before a user can establish an authenticated session. This banner is under complete control of Authorized Administrators in which they specify any warnings regarding unauthorized use of the TOE.

#### 8.2.1.4 O.FAIL_SECURE

*The TOE shall fail in a secure manner following failure of the power-on self tests.*

This TOE Security Objective is satisfied by ensuring that:

- FPT_FLS.1: Requires that on a detected failure the TOE maintains a secure state.

#### 8.2.1.5 O.PROTECTED_COMMUNICATIONS

*The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.*

This TOE Security Objective is satisfied by ensuring that:

- FIA_X8021X_EXT.1 provides the two-way authentication necessary to allow a wireless client access to the wired network, and serves as a part of the 802.11-2007 WPA2 protocol to establish the communication channel with the wireless client.
- FAU_STG_EXT.1: Protects the audit records through transmission between external audit storage.
- FCS_HTTPS_EXT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FCS_IPSEC_EXT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FCS_SSH_EXT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FCS_TLS_EXT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FPT_ITT.1: Requires the TOE provide a mechanism that creates a distinct communication channel between distributed TOE components that protects the data that traverse this channel from disclosure or modification.
- FPT_RPL.1: Ensures that administrator sessions or data communicated with an authorized IT entity cannot be replayed.
- FTP_ITC.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.
- FTP_TRP.1: Requires the TOE provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that protects the data that traverse this channel from disclosure or modification.

#### 8.2.1.6 O.PROTOCOLS

*The TOE will ensure that standardized protocols are implemented in the TOE to RFC and/or Industry specifications to ensure interoperability, that also support communication with a centralized audit server and a RADIUS authentication server.*

This TOE Security Objective is satisfied by ensuring that:

- FCS_HTTPS_EXT.1: References the applicable standards (and indicates any restrictions on those standards) applicable to the protocol they require to be implemented.
- FCS_IPSEC_EXT.1: References the applicable standards (and indicates any restrictions on those standards) applicable to the protocol they require to be implemented.
- FCS_SSH_EXT.1: References the applicable standards (and indicates any restrictions on those standards) applicable to the protocol they require to be implemented.
- FCS_TLS_EXT.1: References the applicable standards (and indicates any restrictions on those standards) applicable to the protocol they require to be implemented.
- FIA_8021X_EXT.1: References the applicable standards (and indicates any restrictions on those standards) applicable to the protocol they require to be implemented.
- FTP_ITC.1: References the applicable standards (and indicates any restrictions on those standards) applicable to the protocol they require to be implemented.

#### 8.2.1.7 O.REPLAY_DETECTION

*The TOE will provide a means to detect and reject the replay of authentication data and other TSF data and security attributes.*

This TOE Security Objective is satisfied by ensuring that:

- FPT_RPL.1: Requires the TOE to detect and reject any attempted replay of authentication data from a remote user.

#### 8.2.1.8 O.RESIDUAL_INFORMATION_CLEARING

*The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.*

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM_EXT.4: Ensures the destruction of any cryptographic keys when no longer needed.
- FDP_RIP.2: is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data. For this TOE it is critical that the memory used to build network packets is either cleared or that some buffer management scheme be employed to prevent the contents of a packet being disclosed in a subsequent packet (e.g., if padding is used in the construction of a packet, it must not contain another user's data or TSF data).

#### 8.2.1.9 O.RESOURCE_AVAILABILITY

*The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (e.g., persistent storage).*

This TOE Security Objective is satisfied by ensuring that:

- FRU_RSA.1: Imposes quotas on exhaustible resources such that resources can be controlled and DoS attacks may be mitigated.

#### 8.2.1.10 O.ROBUST_TOE_ACCESS

*The TOE will provide mechanisms that control an administrator's logical access to the TOE and to control administrative access from a wireless client.*

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: Provides a settable unsuccessful authentication attempt threshold that prevents unauthorized users acting remotely from gaining access to authorized administrator's account by guessing authentication data by locking the targeted account until the Authorized Administrator takes some action (e.g., re-enables the account) or for some Authorized Administrator defined time period.
- FIA_PMG_EXT.1: Defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.
- FIA_UAU.6: Requires a user to re-authenticate when a password is changed or the session is locked.

- FIA_UAU.7: Ensures that authentication feedback is obscured at the local console.
- FIA_UAU_EXT.5: Requires that the TSF provides local authentication methods (one of which is required to be a local password-based mechanism, with other optional (potentially off-box) mechanisms allowed) to ensure that unauthorized users cannot gain logical access to the TOE.
- FIA_UIA_EXT.1: Plays a role in satisfying this objective by ensuring that every user is identified and authenticated before the TOE performs any mediated functions.
- FMT_SMR.1: Controls the administrator's ability to perform administrative actions from a wireless client; the capability must be disabled by default.
- FTA_SSL.3: Takes into account remote sessions. After an Administrator-defined time interval of inactivity remote sessions will be terminated, this includes user proxy sessions and remote administrative sessions. This component is especially necessary since remote sessions are not typically afforded the same physical protections that local sessions are provided.
- FTA_SSL.4: Provides administrators the capability to exit or logoff administrative sessions, rather than wait for the session to be terminated.
- FTA_SSL_EXT.1: Provides the Authenticated Administrator the capability to specify a time interval of inactivity in which an unattended local administrative session would be locked and will require the administrator responsible for that session to re-authenticate before the session can be used to access TOE resources.

#### 8.2.1.11 O.SESSION_LOCK

*The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.*

This TOE Security Objective is satisfied by ensuring that:

- FTA_SSL.3: Takes into account remote sessions. After an Authorized Administrator defined time interval of inactivity remote sessions will be terminated, this includes user proxy sessions and remote administrative sessions. This component is especially necessary because remote sessions are not typically afforded the same physical protections that local sessions are provided.
- FTA_SSL.4: Provides administrators the capability to exit or logoff administrative sessions, rather than wait for the session to be terminated.
- FTA_SSL_EXT.1: Provides an authenticated Administrator the capability to specify a time interval of inactivity in which an unattended local administrative session would be locked and will require the administrator responsible for that session to re-authenticate before the session can be used to access TOE resources.

#### 8.2.1.12 O.SYSTEM_MONITORING

*The TOE will provide the capability to generate audit data and send those data to an external IT entity.*

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: Defines the set of events that the TOE must be capable of recording.
- FAU_GEN.2: Ensures the audit records associate a user identity with the auditable event.
- FAU_SAR.1: Ensures administrators can review the audit records.
- FAU_SAR.2: Ensures only administrators can review the audit records.
- FAU_SEL.1: Allows the administrator to configure which auditable events will be recorded in the audit trail.
- FAU_STG.1: Requires some amount of local audit storage which must be protected from unauthorized access.
- FAU_STG_EXT.1: Protects the audit records through transmission between external audit storage.
- FAU_STG_EXT.3: Defines the set of events that must occur when the link to the external audit storage is not available.
- FPT_STM.1: Requires that the TOE be able to provide reliable time stamps for use in audit records.

#### 8.2.1.13 O.TIME_STAMPS

*The TOE shall provide reliable time stamps and the capability for the administrator to set the time used for these timestamps.*

This TOE Security Objective is satisfied by ensuring that:

- FPT_STM.1: Requires that the TOE be able to provide reliable time stamps for its own use and therefore, partially satisfies this objective. Time stamps include date and time and are reliable in that they are always available to the TOE, and the clock must be monotonically increasing.

#### 8.2.1.14 O.TOE_ADMINISTRATION

*The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.*

This TOE Security Objective is satisfied by ensuring that:

- FIA_PMG_EXT.1: Defines management capabilities and requirements for administrator specification of password/secret strength.
- FIA_UAU_EXT.5: Requires that the TSF provides local authentication methods (one of which is required to be a local password-based mechanism, with other optional (potentially off-box) mechanisms allowed) to ensure that unauthorized users cannot gain logical access to the TOE.
- FMT_MOF.1: Restricts the ability to manage certain functionality and identify security attributes of an authorized administrator.
- FMT_MTD.1(1): Restricts the ability to manage certain functionality and identify security attributes of an authorized administrator.
- FMT_MTD.1(2): Restricts the ability to manage certain functionality and identify security attributes of an authorized administrator.
- FMT_MTD.1(3): Restricts the ability to manage certain functionality and identify security attributes of an authorized administrator.
- FMT_SMF.1: Specifies the management functions that an only administrator must perform.
- FMT_SMR.1: Defines at least one administrator role (Authorized Administrator) to perform administrative actions. The TSF is able to associate a human user to this role.
- FTP_TRP.1: Requires that the TSF provide a trusted path for remote administration.

#### 8.2.1.15 O.TSF_SELF_TEST

*The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.*

This TOE Security Objective is satisfied by ensuring that:

- FPT_FLS.1: Requires that on a detected failure the TOE maintains a secure state.
- FPT_TST_EXT.1: Requires the TOE to provide a suite of self tests to assure the correct operation of the TSF.

#### 8.2.1.16 O.VERIFIABLE_UPDATES

*The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.*

This TOE Security Objective is satisfied by ensuring that:

- FCS_COP.1(2) and FCS_COP.1(3) specify digital signature algorithms and hash functions used in verification of updates.
- FPT_TUD_EXT.1: Provides a way to determine the version of firmware running, initiate an update, and verify the firmware/software updates to the TOE prior to installation.

#### 8.2.1.17 O.WIRELESS_CLIENT_ACCESS

*The TOE will provide the capability to restrict a wireless client in connecting to the TOE.*

This TOE Security Objective is satisfied by ensuring that:

- FTA_TSE.1: Provides the capability to control access by wireless clients based on time of day, their location (e.g., IP address), and other attributes that may be implemented by the TOE.

## 8.3 Security Assurance Requirements Rationale

The Security Assurance Requirements (SARs), which correspond to EAL1, in this ST represent the SARs identified in the WLASPP.

Note that the WLASPP includes a number of 'Assurance Activities' which are in effect refinements of the underlying SARs. As such, those assurance activities have been reproduced in this ST since they need be addressed in the context of the evaluation.

## 8.4 Requirement Dependency Rationale

As can be seen in the following table all of the SFR and SAR dependencies are satisfied in this ST, except where the CC-identified dependency is actually not required as noted twice.

| ST Requirement | CC Dependencies | ST Dependencies |
|---|---|---|
| **FAU_GEN.1** | FPT_STM.1 | FPT_STM.1 |
| **FAU_GEN.2** | FAU_GEN.1 and FIA_UID.1 | FAU_GEN.1 and FIA_UIA_EXT.1 |
| **FAU_SAR.1** | FAU_GEN.1 | FAU_GEN.1 |
| **FAU_SAR.2** | FAU_SAR.1 | FAU_SAR.1 |
| **FAU_SEL.1** | FAU_GEN.1 and FMT_MTD.1 | FAU_GEN.1 and FMT_MTD.1(1) |
| **FAU_STG.1** | FAU_GEN.1 | FAU_GEN.1 |
| **FAU_STG_EXT.1** | FAU_GEN.1 | FAU_GEN.1 |
| **FAU_STG_EXT.3** | FAU_STG_EXT.1 | FAU_STG_EXT.1 |
| **FCS_CKM.1(1)** | none | none |
| **FCS_CKM.1(2)** | none | none |
| **FCS_CKM.2(1)** | none | none |
| **FCS_CKM.2(2)** | none | none |
| **FCS_CKM_EXT.4** | (FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) | FCS_CKM.1(1) |
| **FCS_COP.1(1)** | (FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 | FCS_CKM.1(1) and FCS_CKM_EXT.4 |
| **FCS_COP.1(2)** | (FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 | FCS_CKM.1(2) and FCS_CKM_EXT.4 |
| **FCS_COP.1(3)** | (FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 | ***No keys required for hashing.** |
| **FCS_COP.1(4)** | (FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 | ***No keys required for hashing.** |
| **FCS_COP.1(5)** | (FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 | FCS_CKM.1(1) and FCS_CKM_EXT.4 |
| **FCS_HTTPS_EXT.1** | FCS_TLS_EXT.1 | FCS_TLS_EXT.1 |
| **FCS_IPSEC_EXT.1** | FCS_COP.1 | FCS_COP.1(1) |
| **FCS_RBG_EXT.1** | none | none |
| **FCS_SSH_EXT.1** | FCS_COP.1 | FCS_COP.1(1) |
| **FCS_TLS_EXT.1** | FCS_COP.1 | FCS_COP.1(1) |
| **FDP_RIP.2** | none | none |
| **FIA_8021X_EXT.1** | none | none |
| **FIA_AFL.1** | FIA_UAU.1 | FIA_UIA_EXT.1 |
| **FIA_PMG_EXT.1** | none | none |
| **FIA_PSK_EXT.1** | none | none |
| **FIA_UAU.6** | none | none |
| **FIA_UAU.7** | FIA_UAU.1 | FIA_UIA_EXT.1 |
| **FIA_UAU_EXT.5** | none | none |
| **FIA_UIA_EXT.1** | none | none |
| **FIA_X509_EXT.1** | none | none |
| **FMT_MOF.1** | FMT_SMR.1 and FMT_SMF.1 | FMT_SMR.1 and FMT_SMF.1 |
| **FMT_MTD.1(1)** | none | none |
| **FMT_MTD.1(2)** | none | none |
| **FMT_MTD.1(3)** | none | none |
| **FMT_SMF.1** | none | none |
| **FMT_SMR.1** | FIA_UID.1 | FIA_UIA_EXT.1 |

| ST Requirement | CC Dependencies | ST Dependencies |
|---|---|---|
| **FPT_FLS.1** | none | none |
| **FPT_ITT.1** | none | none |
| **FPT_RPL.1** | none | none |
| **FPT_STM.1** | none | none |
| **FPT_TST_EXT.1** | none | none |
| **FPT_TUD_EXT.1** | none | none |
| **FRU_RSA.1** | none | none |
| **FTA_SSL.3** | none | none |
| **FTA_SSL.4** | none | none |
| **FTA_SSL_EXT.1** | none | none |
| **FTA_TAB.1** | none | none |
| **FTA_TSE.1** | none | none |
| **FTP_ITC.1** | none | none |
| **FTP_TRP.1** | none | none |
| **ADV_FSP.1** | none | none |
| **AGD_OPE.1** | ADV_FSP.1 | ADV_FSP.1 |
| **AGD_PRE.1** | none | none |
| **ALC_CMC.1** | ALC_CMS.1 | ALC_CMS.1 |
| **ALC_CMS.1** | none | none |
| **ATE_IND.1** | ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1 | ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1 |
| **AVA_VAN.1** | ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1 | ADV_FSP.1 and AGD_OPE.1 and AGD_PRE.1 |

**Table 9 Requirement Dependencies**

## 8.5 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 10 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

| | Security audit | Cryptographic support | User data protection | Identification and authentication | Security management | Protection of the TSF | Resource utilisation | TOE access | Trusted path/channels |
|---|---|---|---|---|---|---|---|---|---|
| **FAU_GEN.1** | X | | | | | | | | |
| **FAU_GEN.2** | X | | | | | | | | |
| **FAU_SAR.1** | X | | | | | | | | |
| **FAU_SAR.2** | X | | | | | | | | |
| **FAU_SEL.1** | X | | | | | | | | |
| **FAU_STG.1** | X | | | | | | | | |
| **FAU_STG_EXT.1** | X | | | | | | | | |
| **FAU_STG_EXT.3** | X | | | | | | | | |
| **FCS_CKM.1(1)** | | X | | | | | | | |
| **FCS_CKM.1(2)** | | X | | | | | | | |

| | Security audit | Cryptographic support | User data protection | Identification and authentication | Security management | Protection of the TSF | Resource utilisation | TOE access | Trusted path/channels |
|---|---|---|---|---|---|---|---|---|---|
| **FCS_CKM.2(1)** | | X | | | | | | | |
| **FCS_CKM.2(2)** | | X | | | | | | | |
| **FCS_CKM_EXT.4** | | X | | | | | | | |
| **FCS_COP.1(1)** | | X | | | | | | | |
| **FCS_COP.1(2)** | | X | | | | | | | |
| **FCS_COP.1(3)** | | X | | | | | | | |
| **FCS_COP.1(4)** | | X | | | | | | | |
| **FCS_COP.1(5)** | | X | | | | | | | |
| **FCS_HTTPS_EXT.1** | | X | | | | | | | |
| **FCS_IPSEC_EXT.1** | | X | | | | | | | |
| **FCS_RBG_EXT.1** | | X | | | | | | | |
| **FCS_SSH_EXT.1** | | X | | | | | | | |
| **FCS_TLS_EXT.1** | | X | | | | | | | |
| **FDP_RIP.2** | | | X | | | | | | |
| **FIA_8021X_EXT.1** | | | | X | | | | | |
| **FIA_AFL.1** | | | | X | | | | | |
| **FIA_PMG_EXT.1** | | | | X | | | | | |
| **FIA_PSK_EXT.1** | | | | X | | | | | |
| **FIA_UAU.6** | | | | X | | | | | |
| **FIA_UAU.7** | | | | X | | | | | |
| **FIA_UAU_EXT.5** | | | | X | | | | | |
| **FIA_UIA_EXT.1** | | | | X | | | | | |
| **FIA_X509_EXT.1** | | | | X | | | | | |
| **FMT_MOF.1** | | | | | X | | | | |
| **FMT_MTD.1(1)** | | | | | X | | | | |
| **FMT_MTD.1(2)** | | | | | X | | | | |
| **FMT_MTD.1(3)** | | | | | X | | | | |
| **FMT_SMF.1** | | | | | X | | | | |
| **FMT_SMR.1** | | | | | X | | | | |
| **FPT_FLS.1** | | | | | | X | | | |
| **FPT_ITT.1** | | | | | | X | | | |
| **FPT_RPL.1** | | | | | | X | | | |
| **FPT_STM.1** | | | | | | X | | | |
| **FPT_TST_EXT.1** | | | | | | X | | | |
| **FPT_TUD_EXT.1** | | | | | | X | | | |
| **FRU_RSA.1** | | | | | | | X | | |
| **FTA_SSL.3** | | | | | | | | X | |
| **FTA_SSL.4** | | | | | | | | X | |
| **FTA_SSL_EXT.1** | | | | | | | | X | |
| **FTA_TAB.1** | | | | | | | | X | |
| **FTA_TSE.1** | | | | | | | | X | |
| **FTP_ITC.1** | | | | | | | | | X |
| **FTP_TRP.1** | | | | | | | | | X |

**Table 10 Security Functions vs. Requirements Mapping**